

FIG. 1

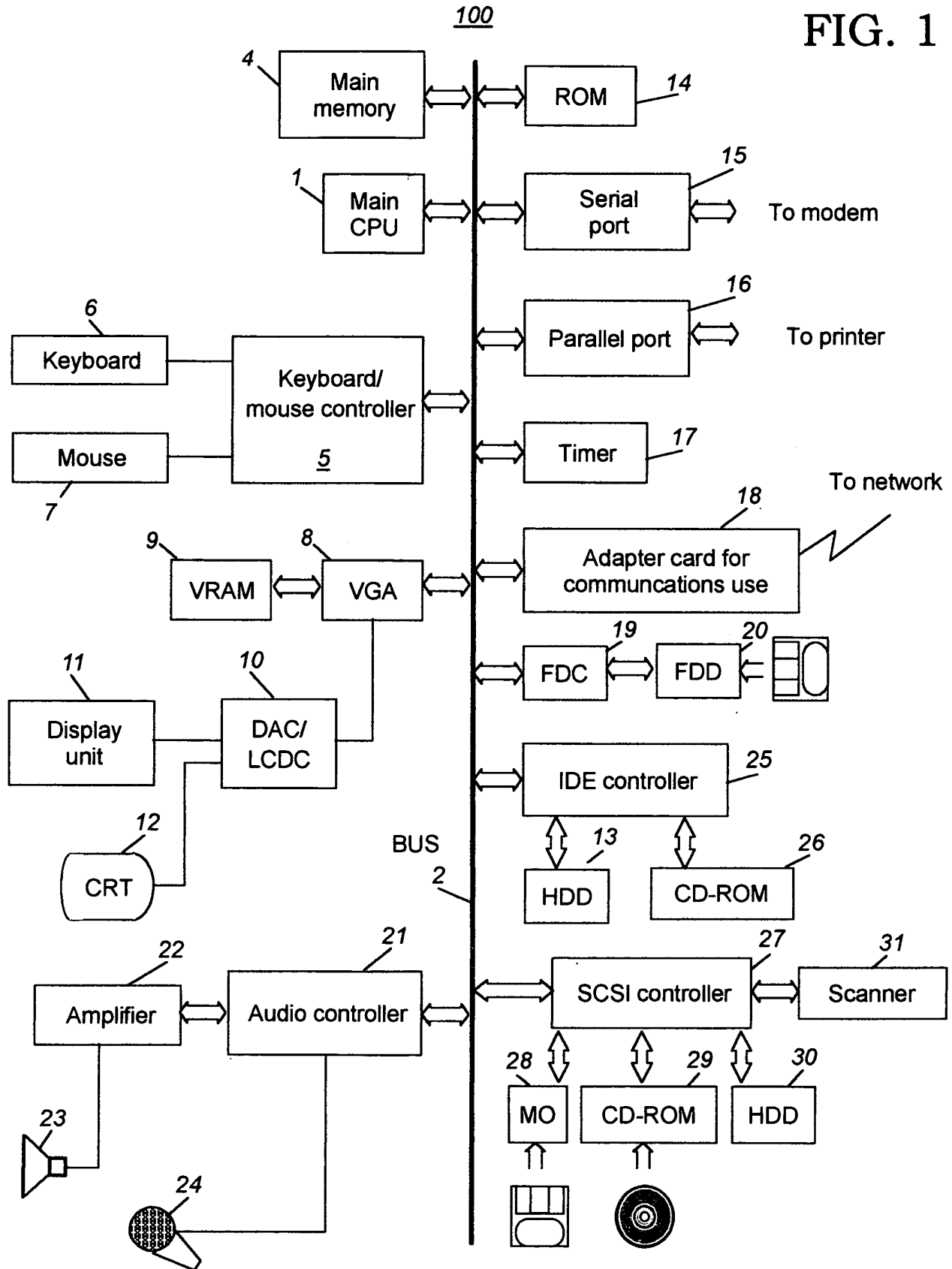


FIG. 2A

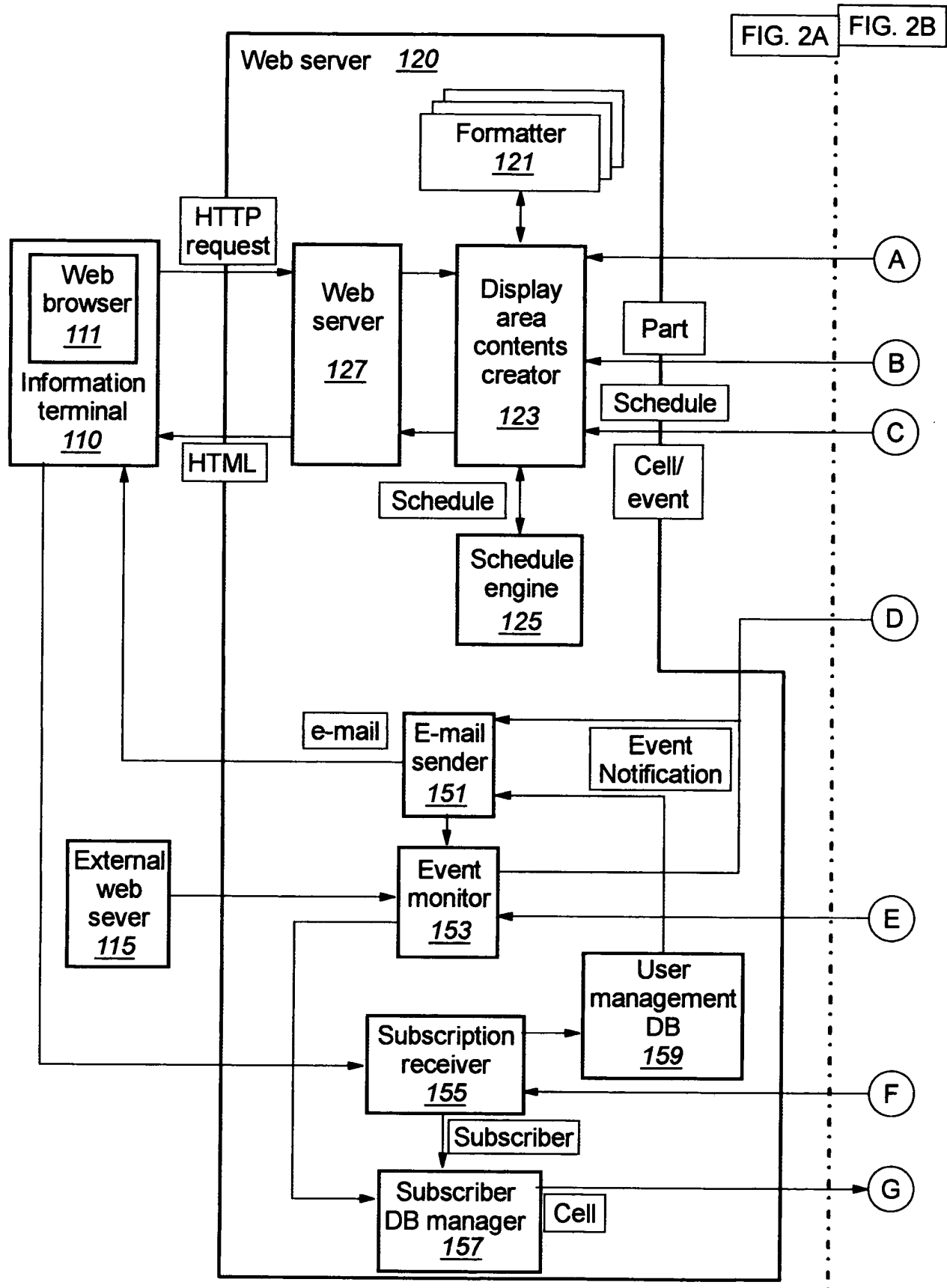


FIG. 2B

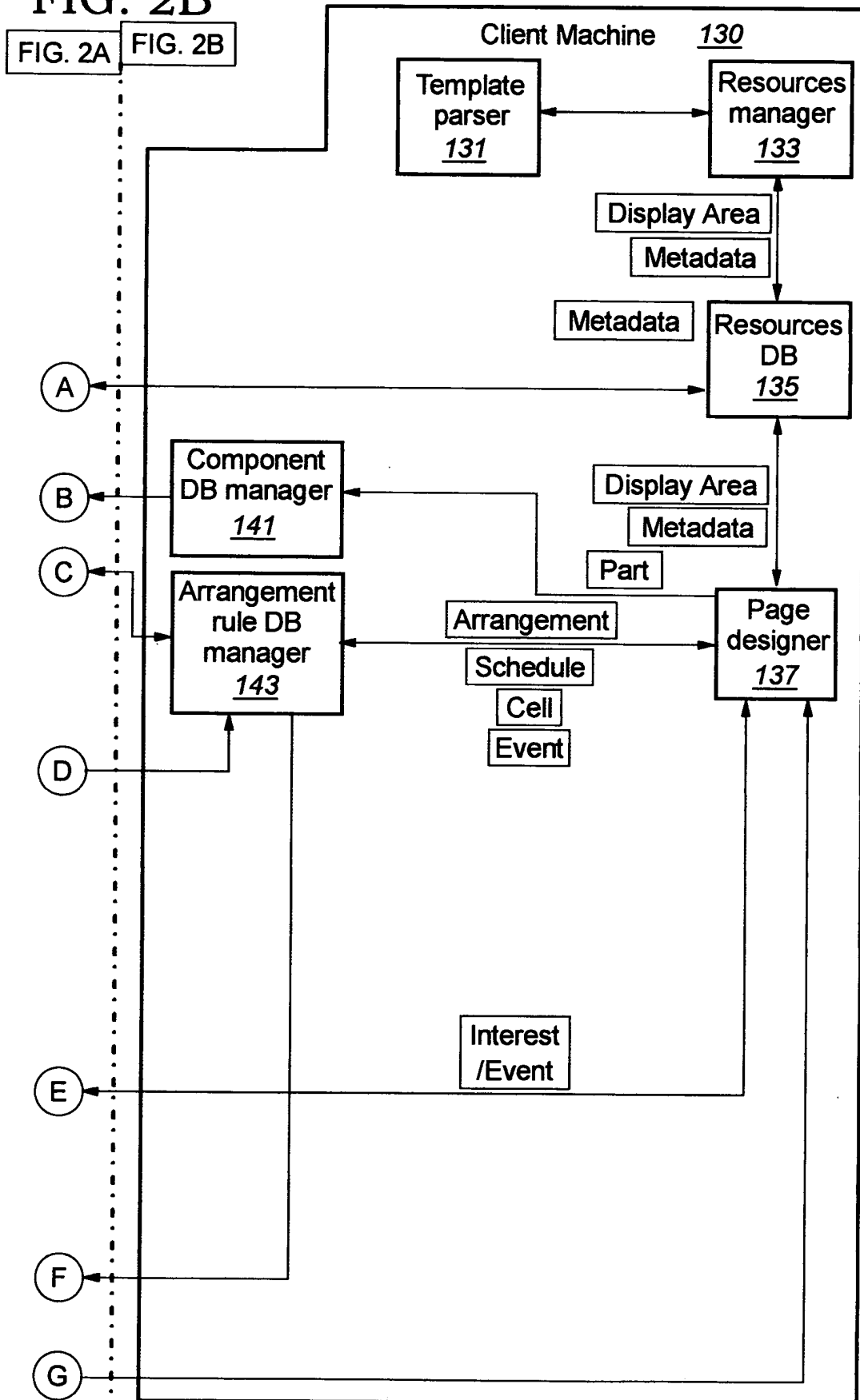


FIG. 3

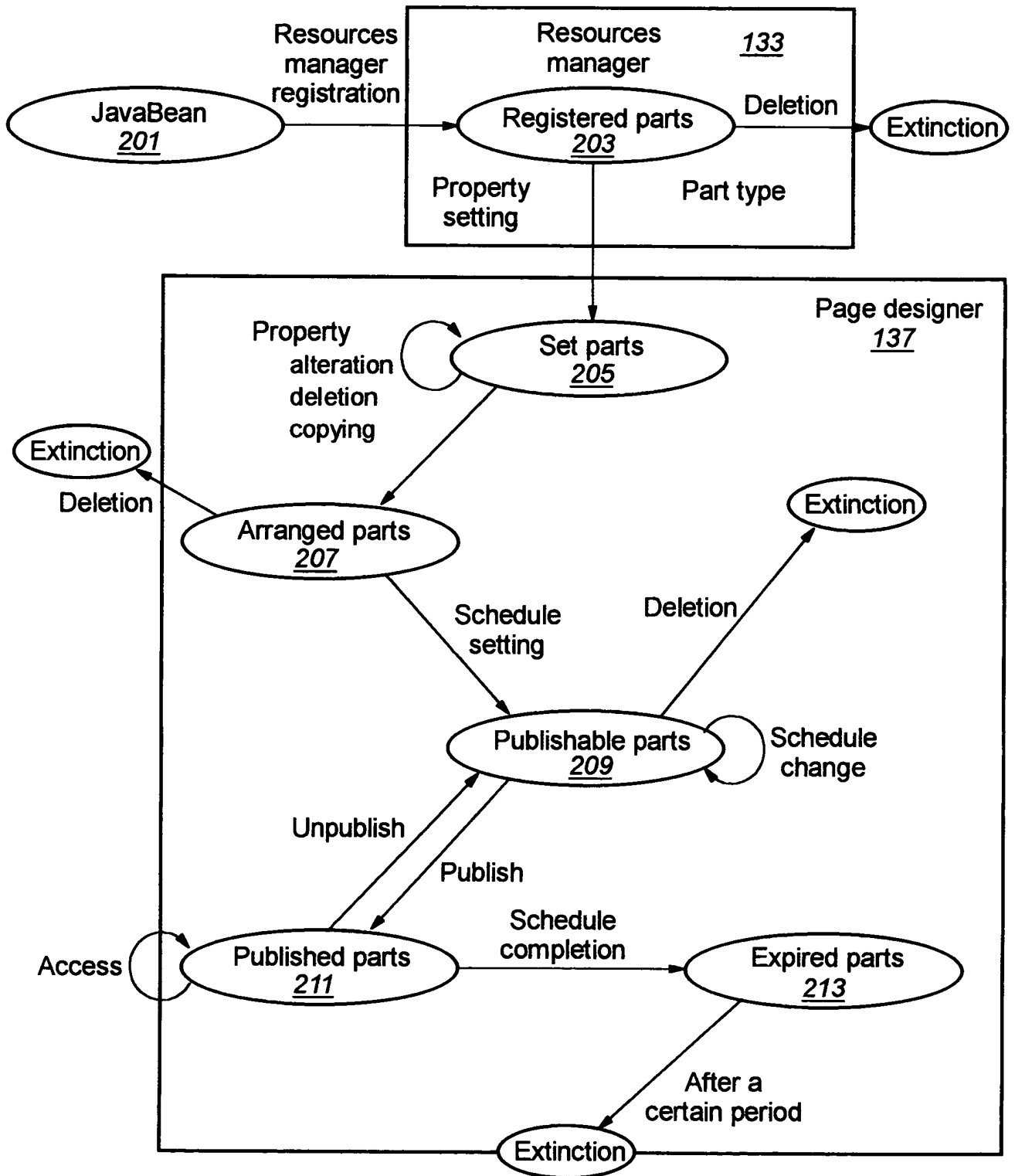


FIG. 4

Contents creator

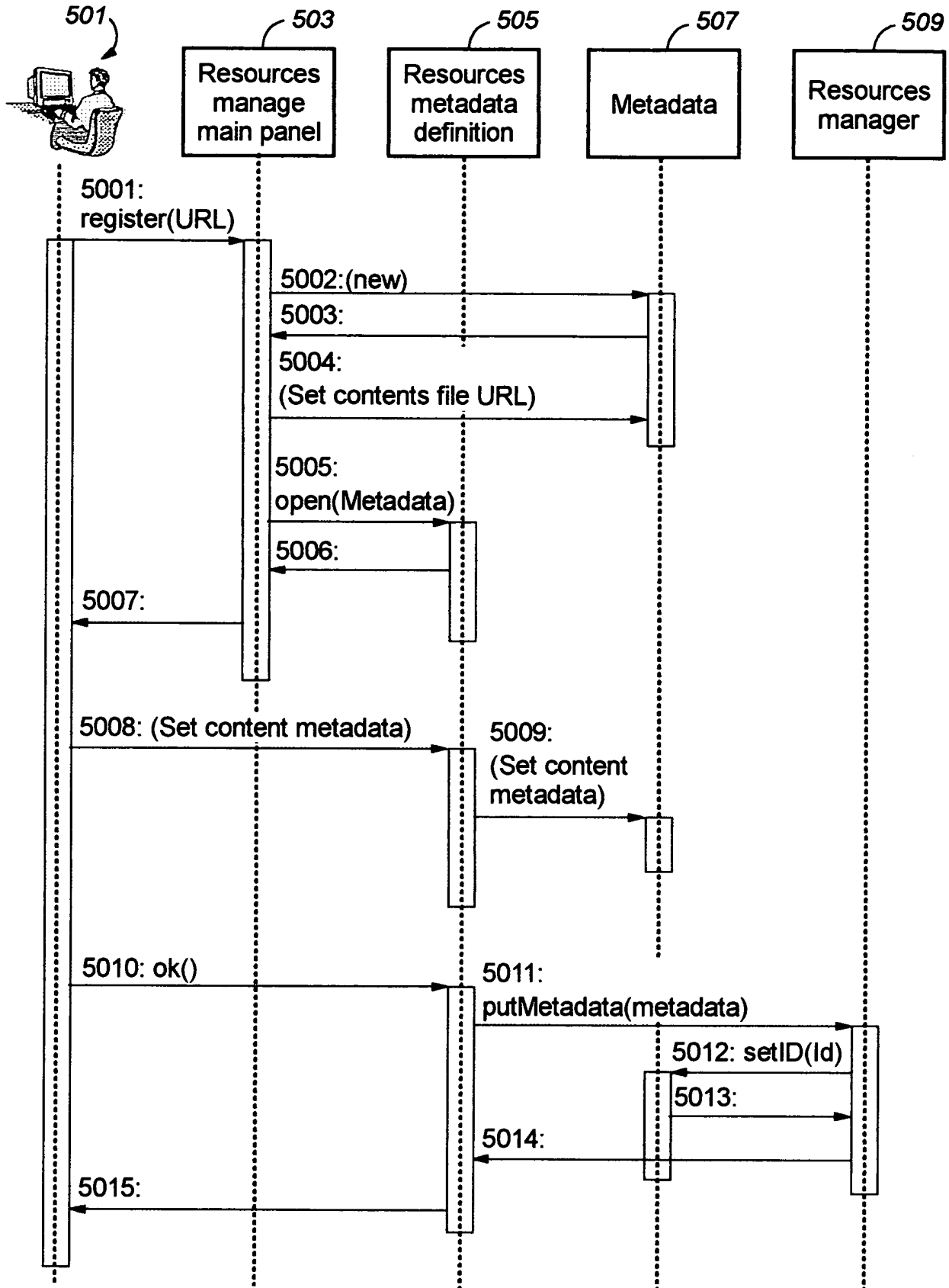


FIG. 5

Contents creator

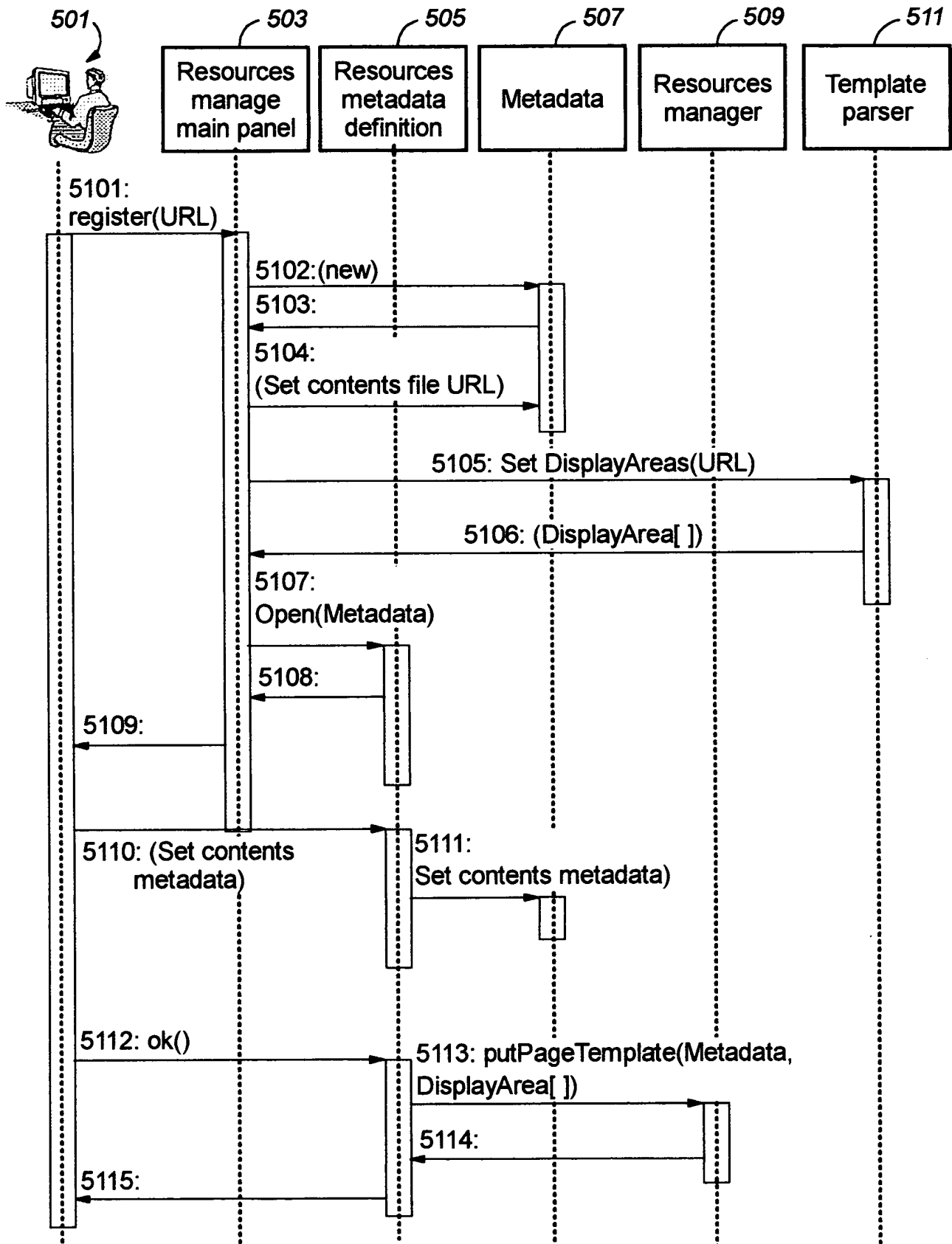
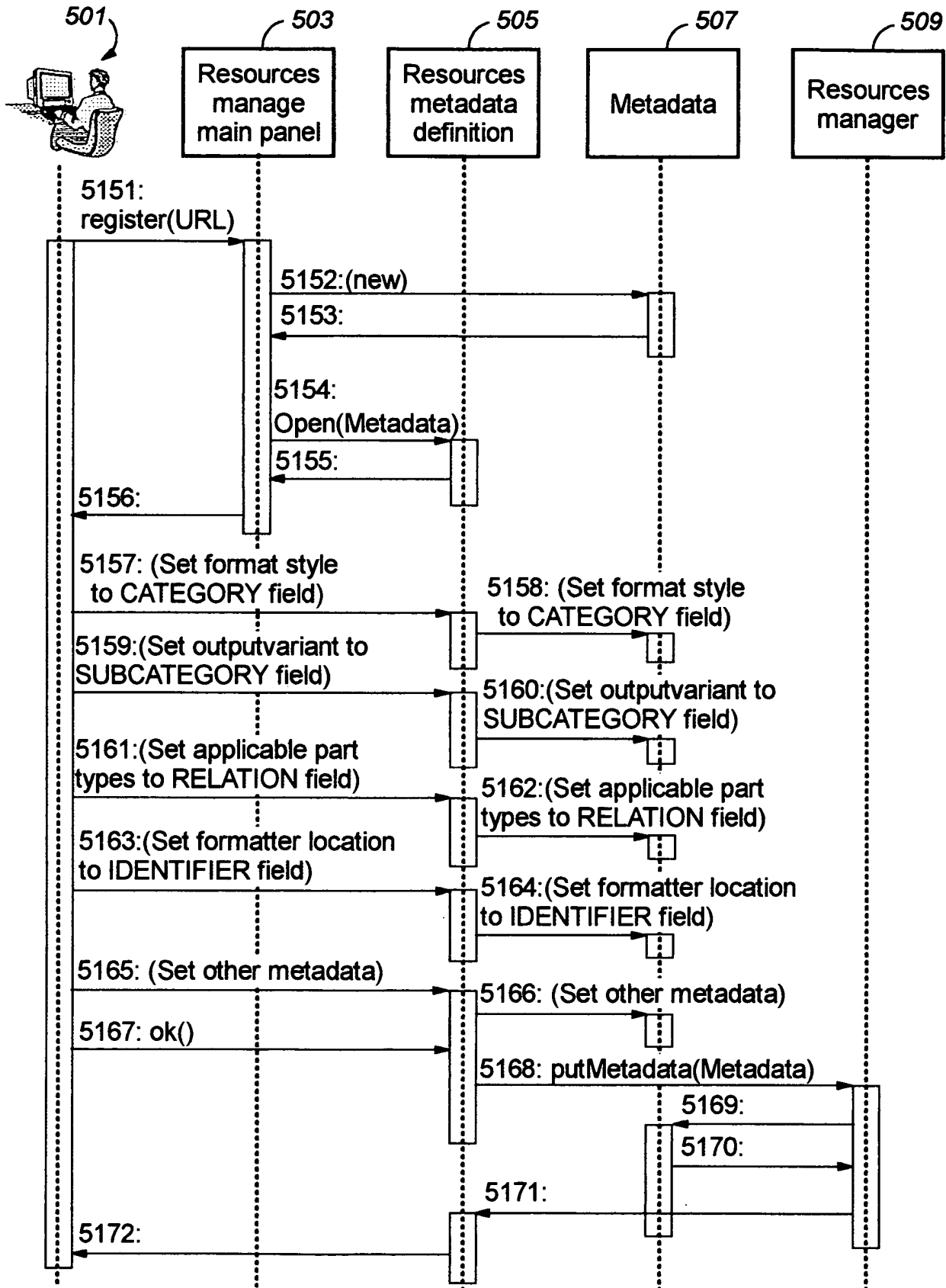


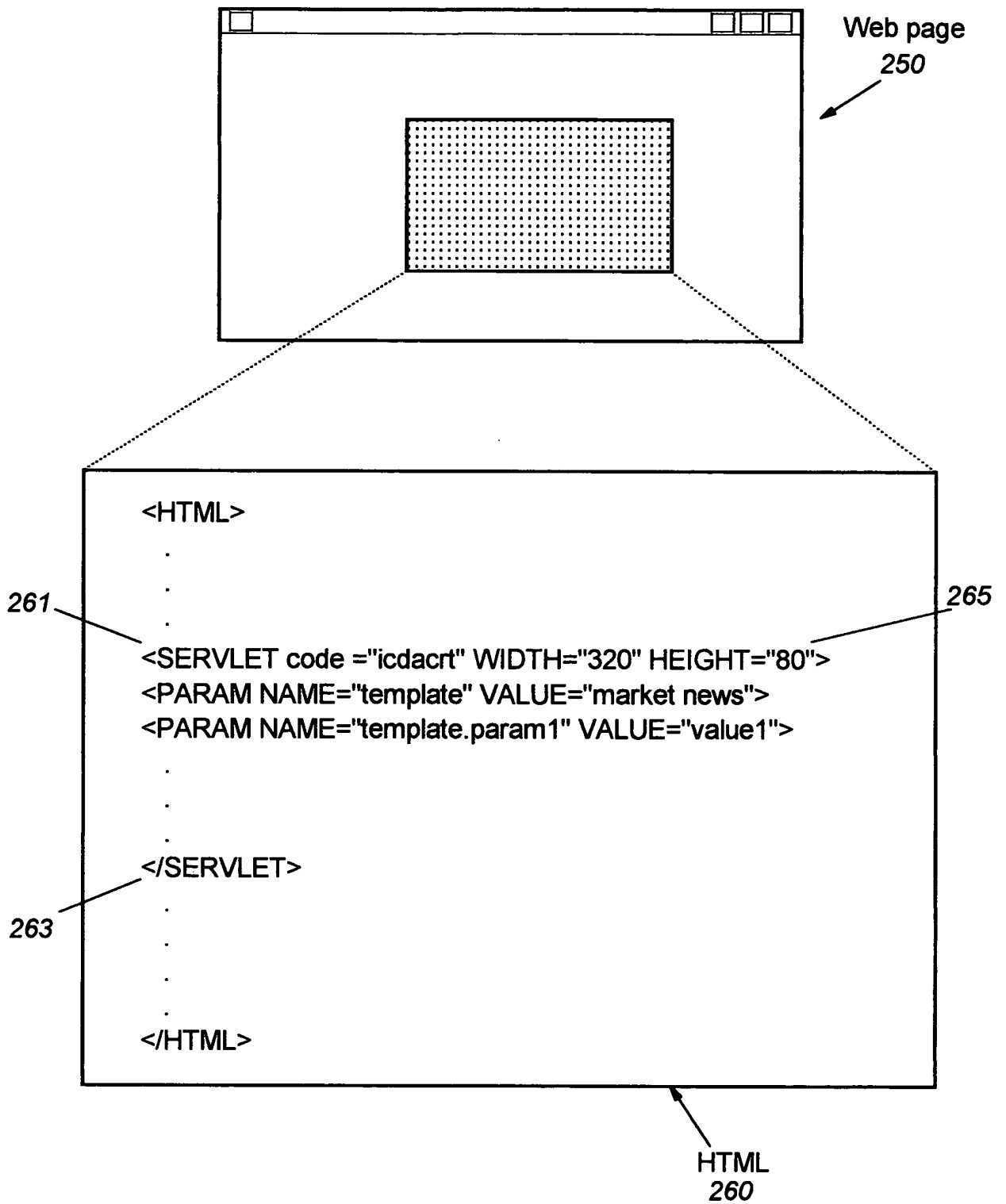
FIG. 6

Contents creator



00000000-00000000

FIG. 7



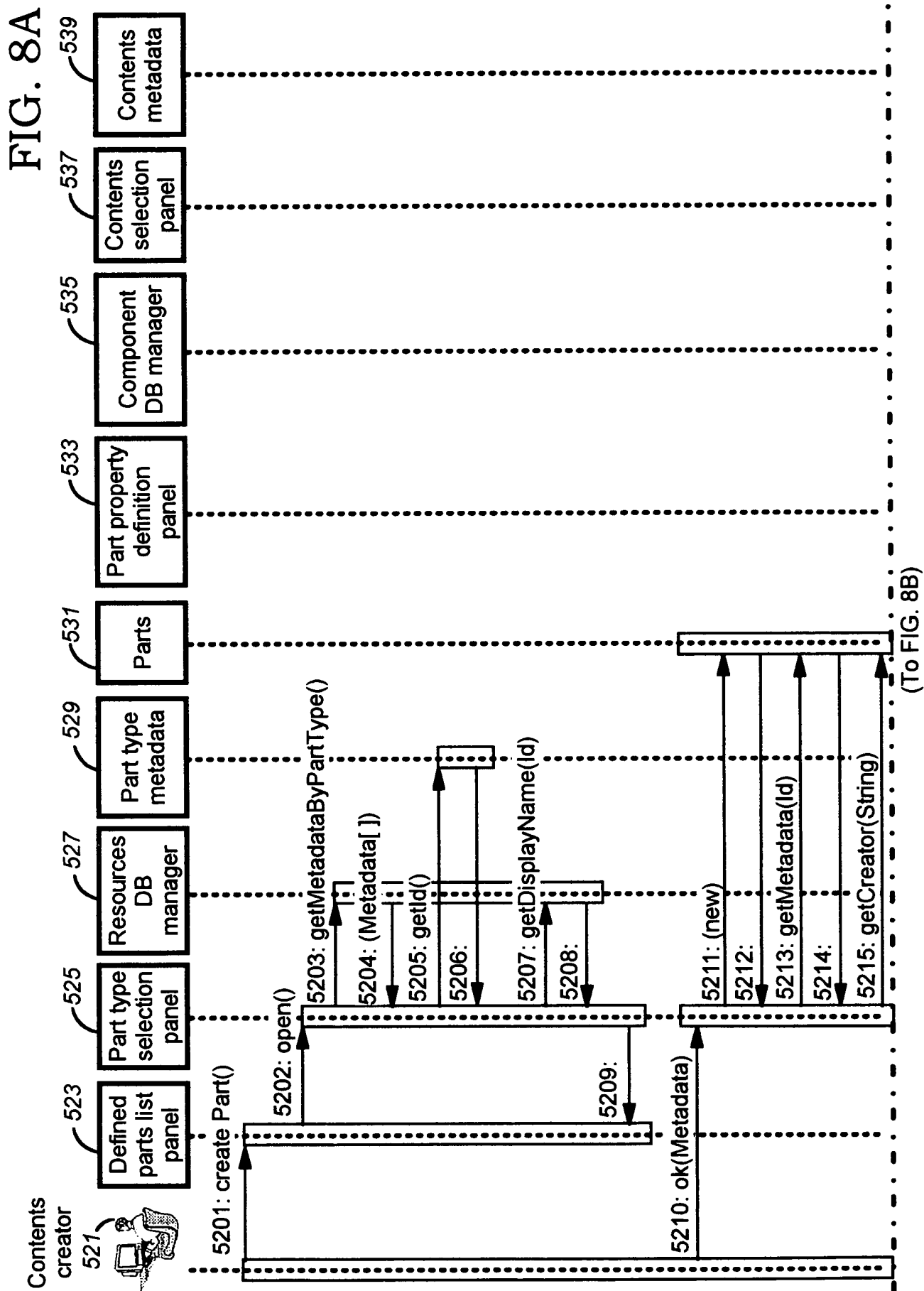
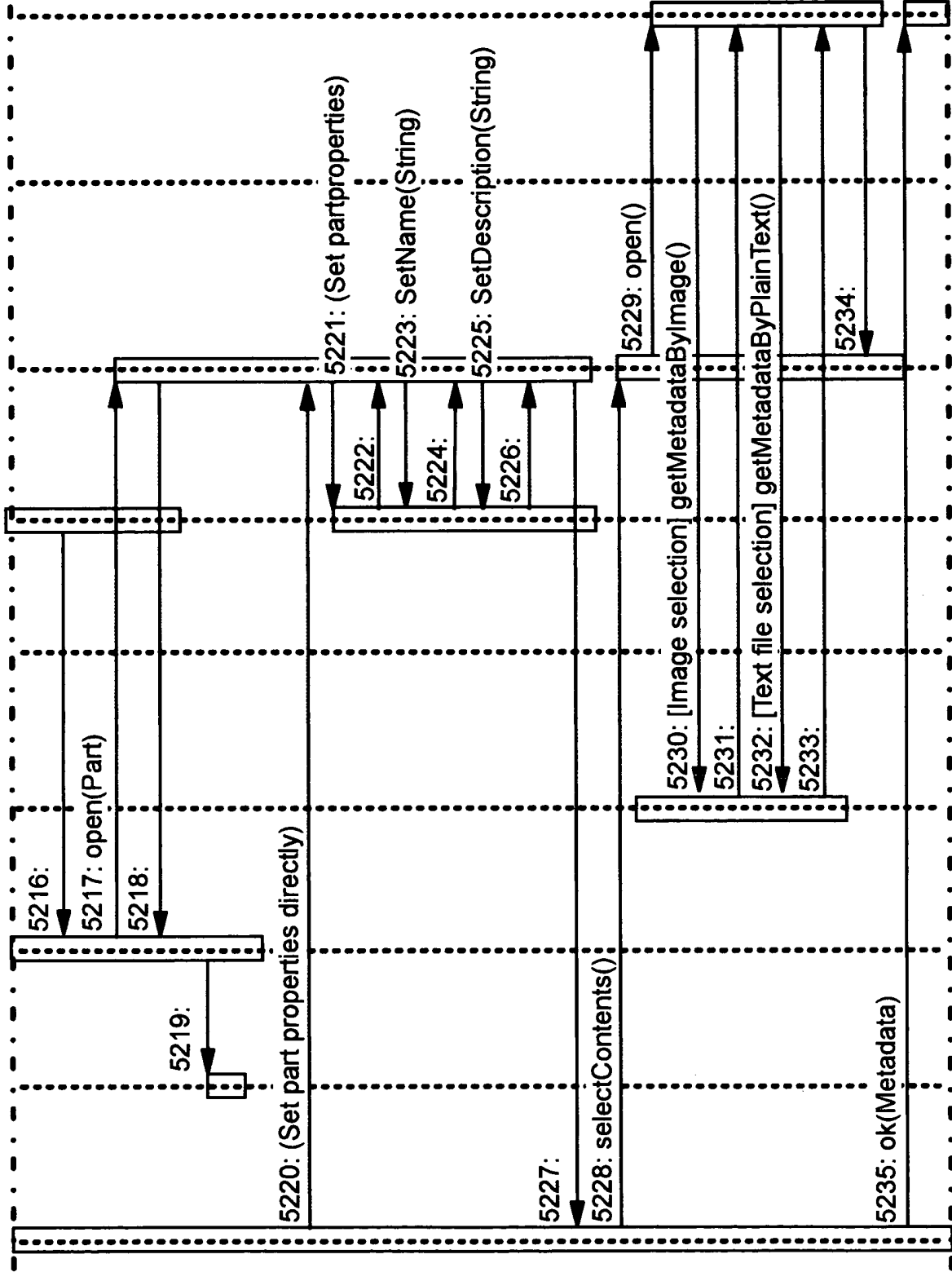


FIG. 8B

(From FIG. 8A)



(To FIG. 9)

FIG. 9

(From FIG. 8B)

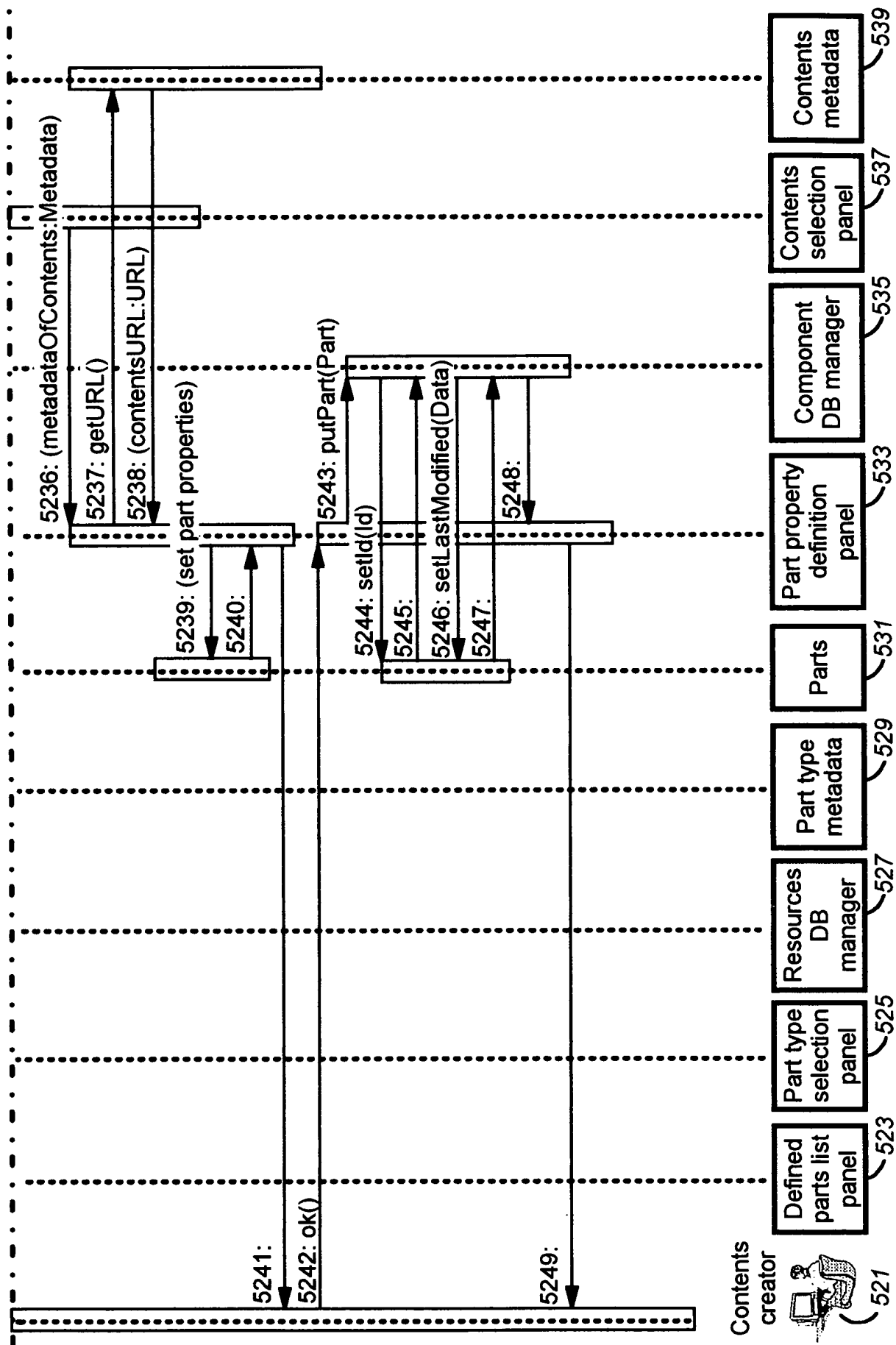


FIG. 10

```
graph TD
    Start((Start)) --> 601
    601 --> End((End))
    601 -- "Parts management" --> 603
    603 -- "Close" --> 601
    603 -- "Copy" --> 603
    603 -- "Create" --> 605
    605 -- "Cancel" --> 603
    605 -- "OK" --> 607
    603 -- "Delete" --> 609
    603 -- "OK/cancel" --> 607
    607 -- "Edit" --> 603
    601 -- "Arranged parts list" --> 611
    611 -- "Close" --> 601
    601 -- "Parts arrangement /schedule" --> 613
    613 -- "Close" --> 601
    613 -- "Schedule" --> 615
    615 -- "Close" --> 613
    613 -- "Parts arrangement" --> 617
    617 -- "Close" --> 613
    617 -- "Schedule" --> 619
    619 -- "Close" --> 617
    617 -- "Publish/unpublish" --> 617
    617 -- "Add" --> 621
    621 -- "OK/cancel" --> 617
    617 -- "Delete" --> 623
    623 -- "OK/cancel" --> 617
```

The flowchart illustrates the sequence of screens and user interactions in a page design system. The screens are represented by ovals and the interactions by arrows with labels.

- Page designer main screen 601**: The starting point, with a "Start" arrow pointing to it and an "End" arrow pointing away from it. It has a "Parts management" arrow to screen 603 and a "Close" arrow back to itself.
- Set parts list screen 603**: Reached from 601. It has a "Copy" self-loop, a "Create" arrow to screen 605, a "Cancel" arrow back to 603, a "Delete" arrow to screen 609, and "OK/cancel" arrows to screen 607.
- Part type selection screen 605**: Reached from 603. It has an "OK" arrow to screen 607.
- Part properties 607**: Reached from 605. It has an "Edit" arrow back to screen 603.
- Confirmation screen 609**: Reached from 603.
- Set parts list screen 611**: Reached from 601 via the "Arranged parts list" arrow. It has a "Close" arrow back to 601.
- Display area list screen 613**: Reached from 601 via the "Parts arrangement /schedule" arrow. It has a "Close" arrow back to 601, a "Schedule" arrow to screen 615, and a "Parts arrangement" arrow to screen 617.
- Page schedule screen 615**: Reached from 613. It has a "Close" arrow back to 613.
- Parts arrangement screen 617**: Reached from 613. It has a "Close" arrow back to 613, a "Schedule" arrow to screen 619, a "Publish/unpublish" self-loop, an "Add" arrow to screen 621, and a "Delete" arrow to screen 623.
- Page schedule definition screen 619**: Reached from 617. It has a "Close" arrow back to 617.
- Part selection screen 621**: Reached from 617. It has an "OK/cancel" arrow back to 617.
- Confirmation screen 623**: Reached from 617. It has an "OK/cancel" arrow back to 617.

FIG. 11B

(From FIG. 11A)

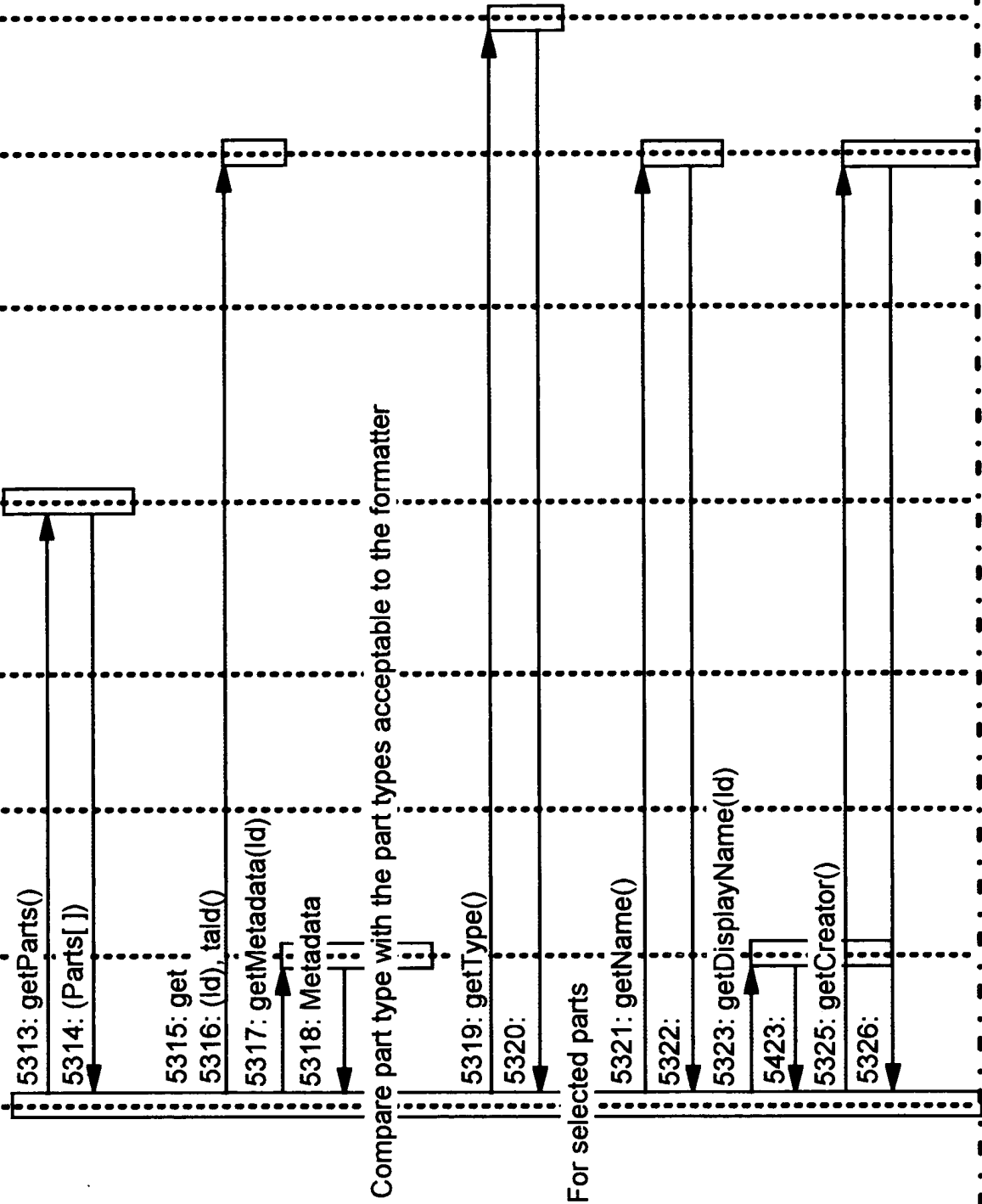


FIG. 12

(From FIG. 11B)

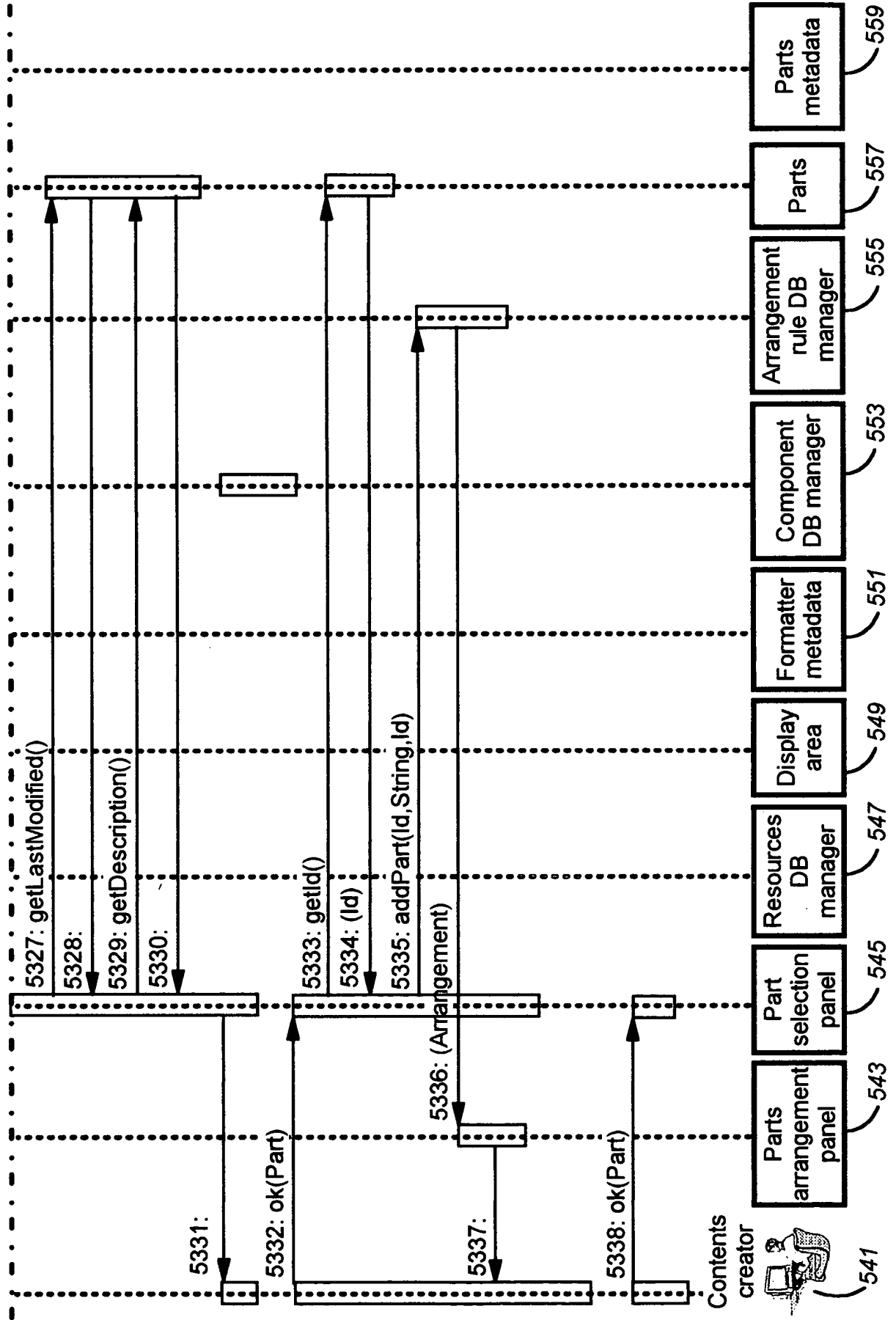


FIG. 13

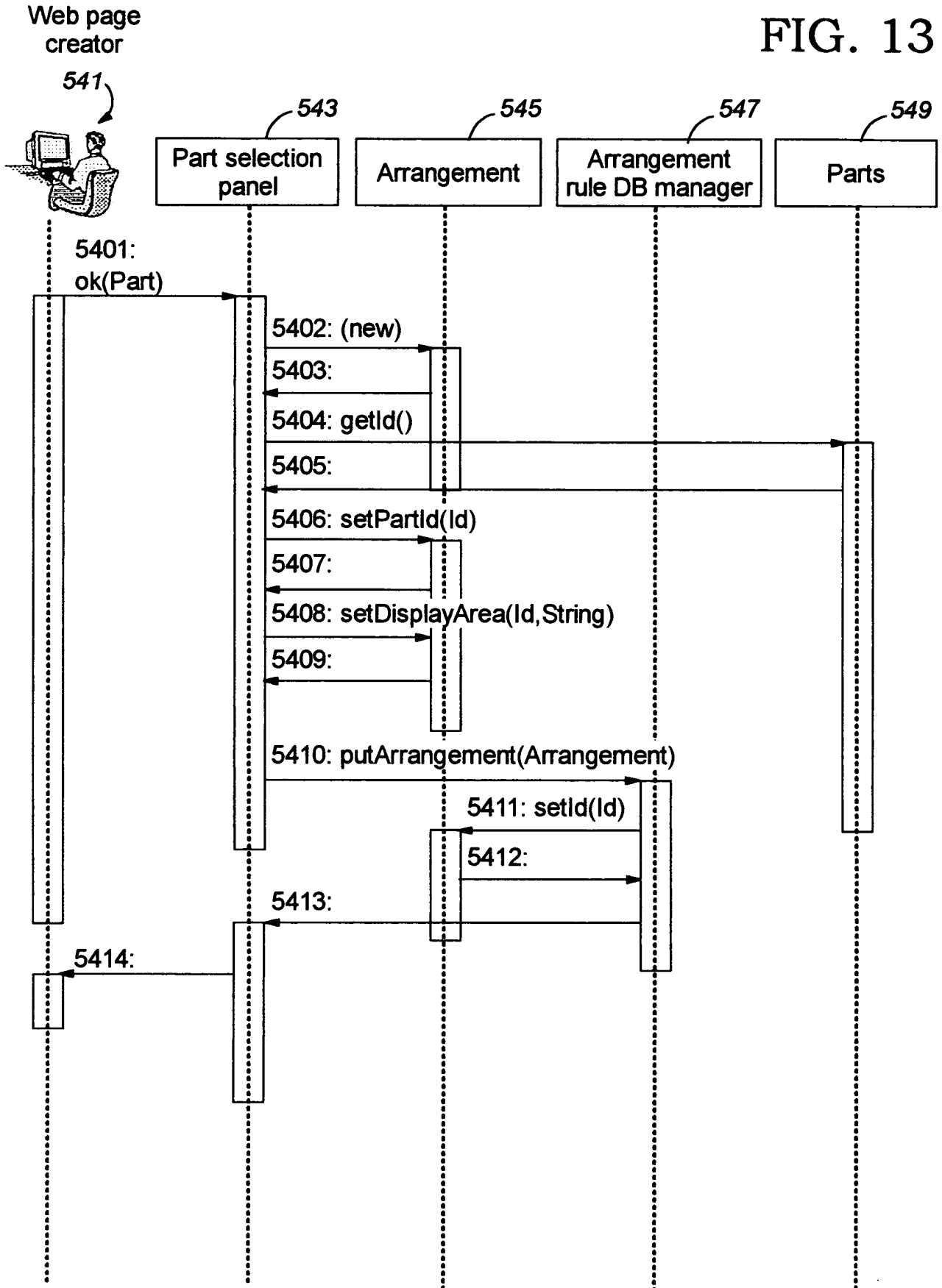


FIG. 14

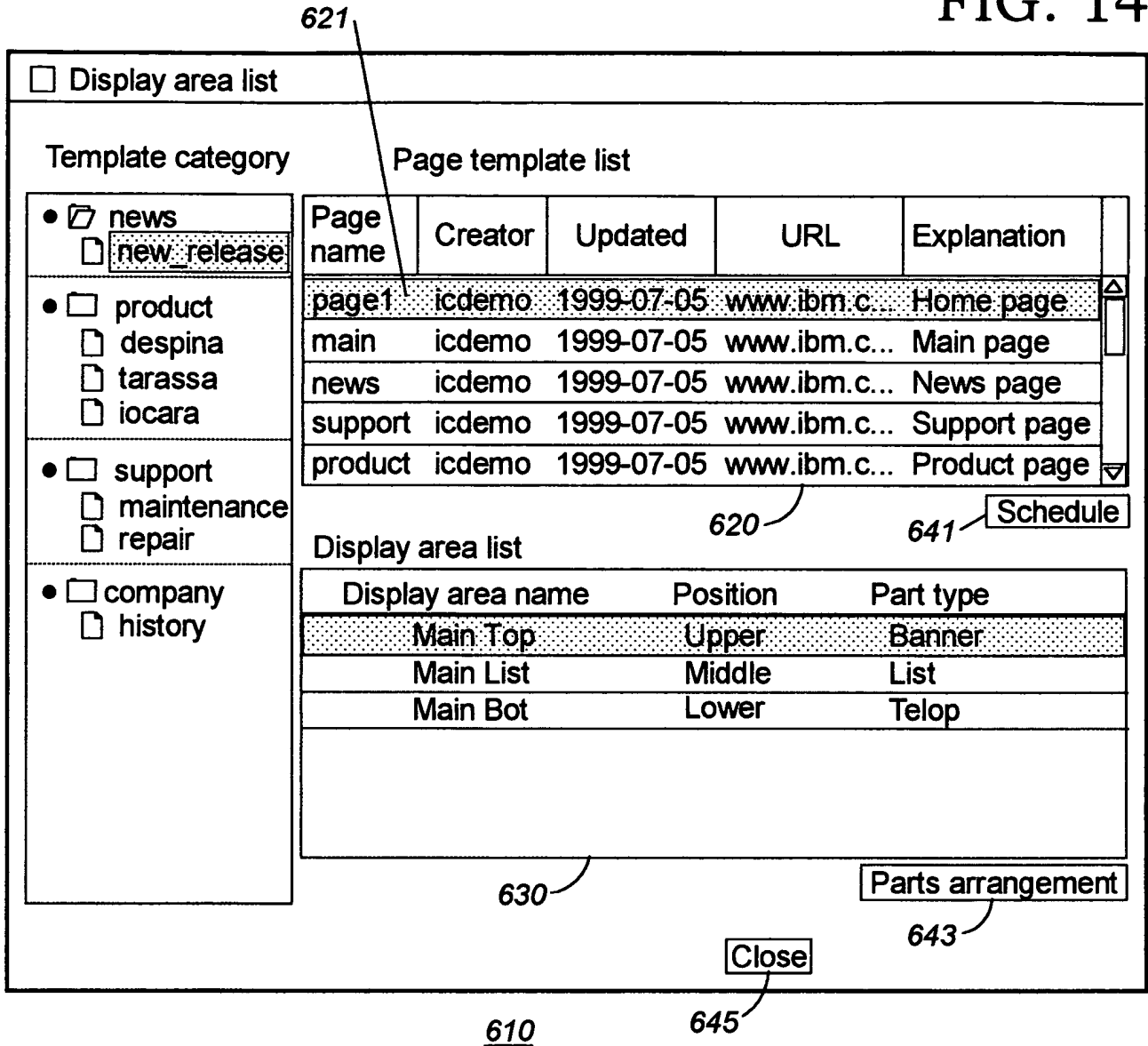


FIG. 16

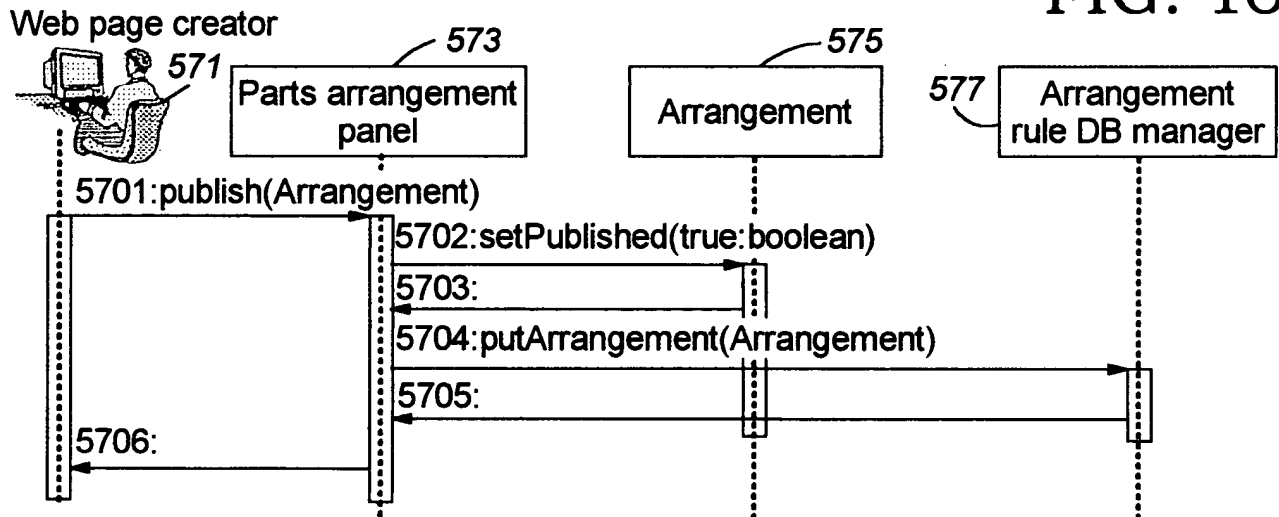


FIG. 15

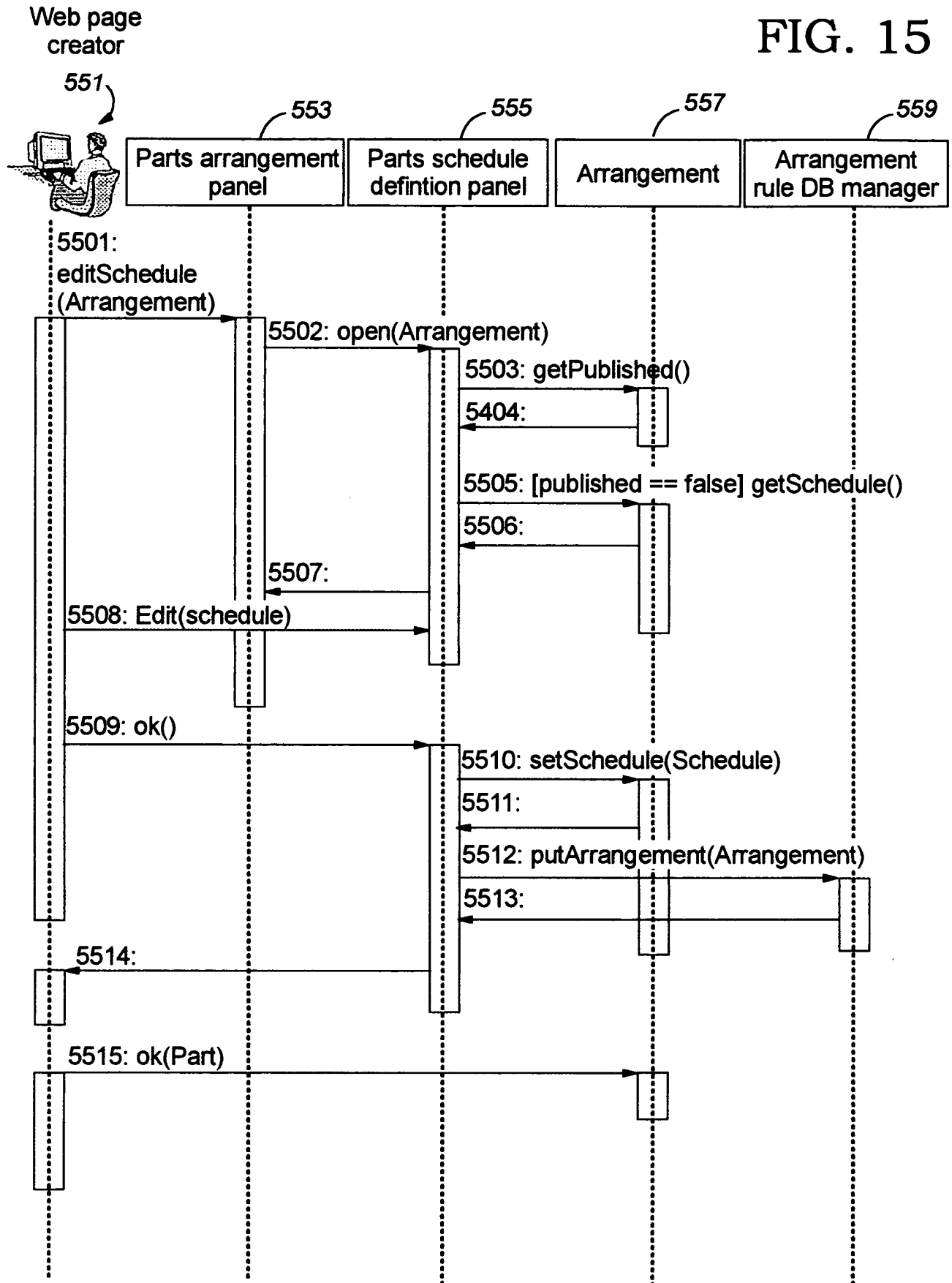


FIG. 17A

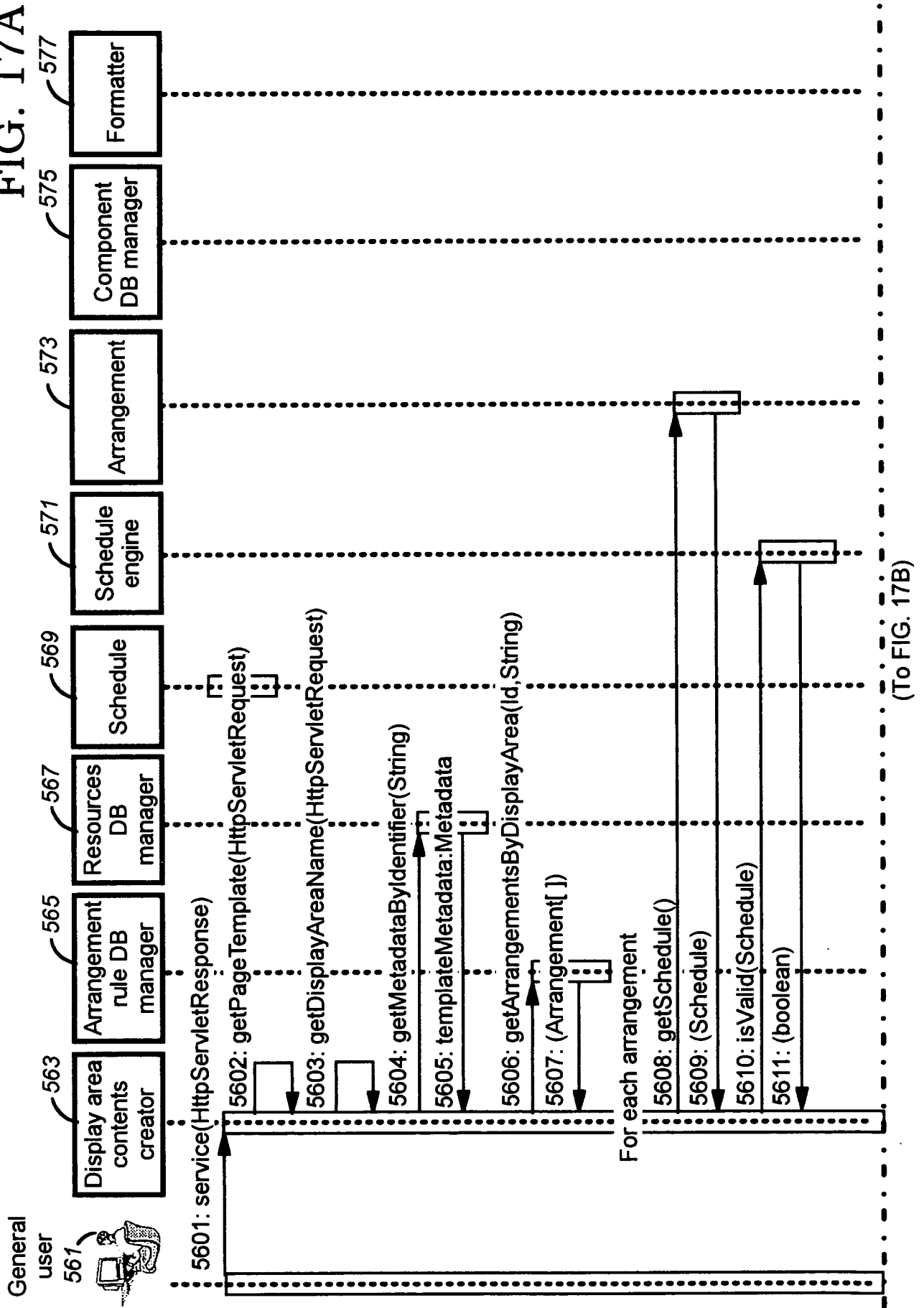


FIG. 17B

(From FIG. 17A)

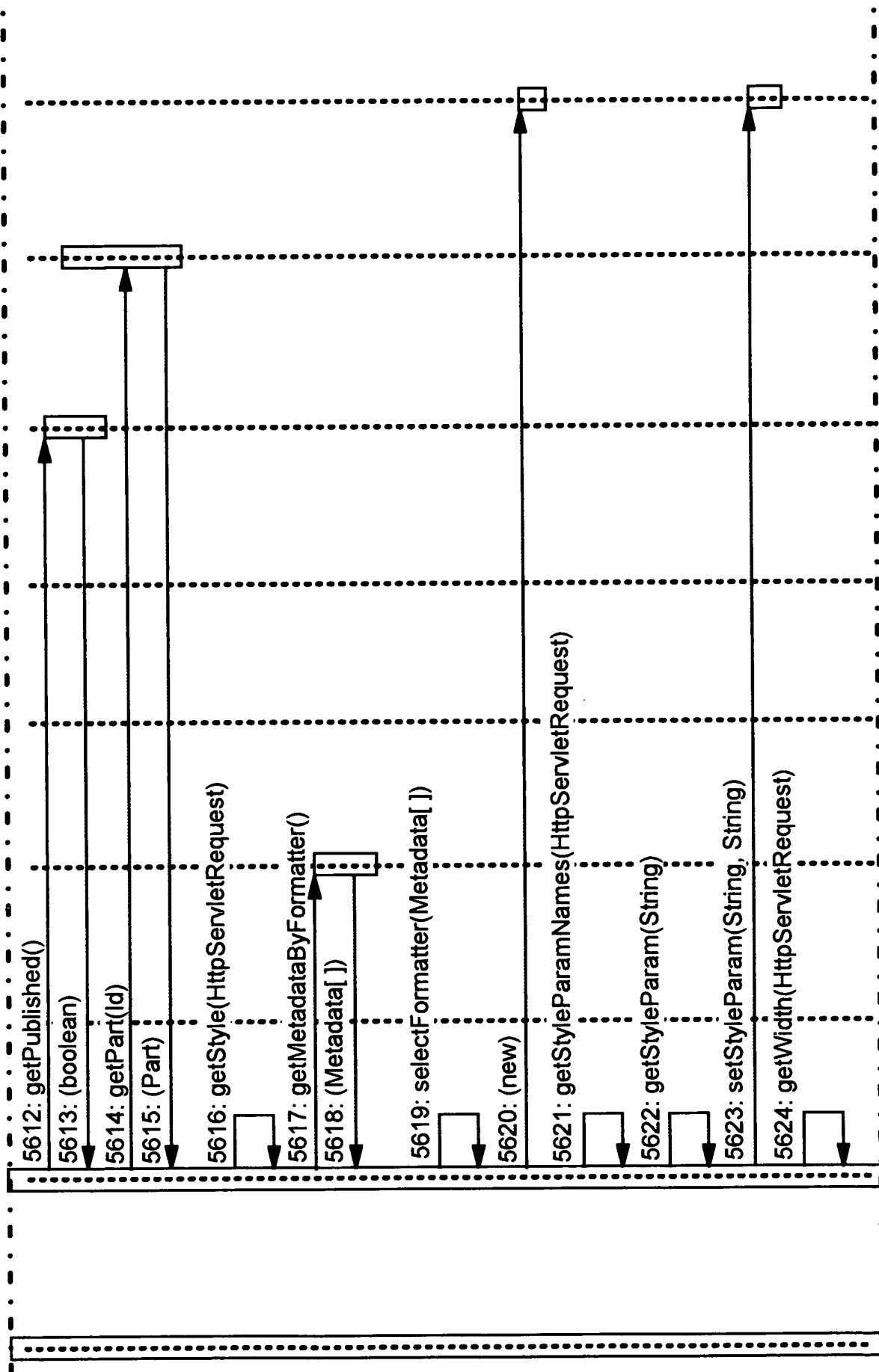


FIG. 18

(From FIG. 17B)

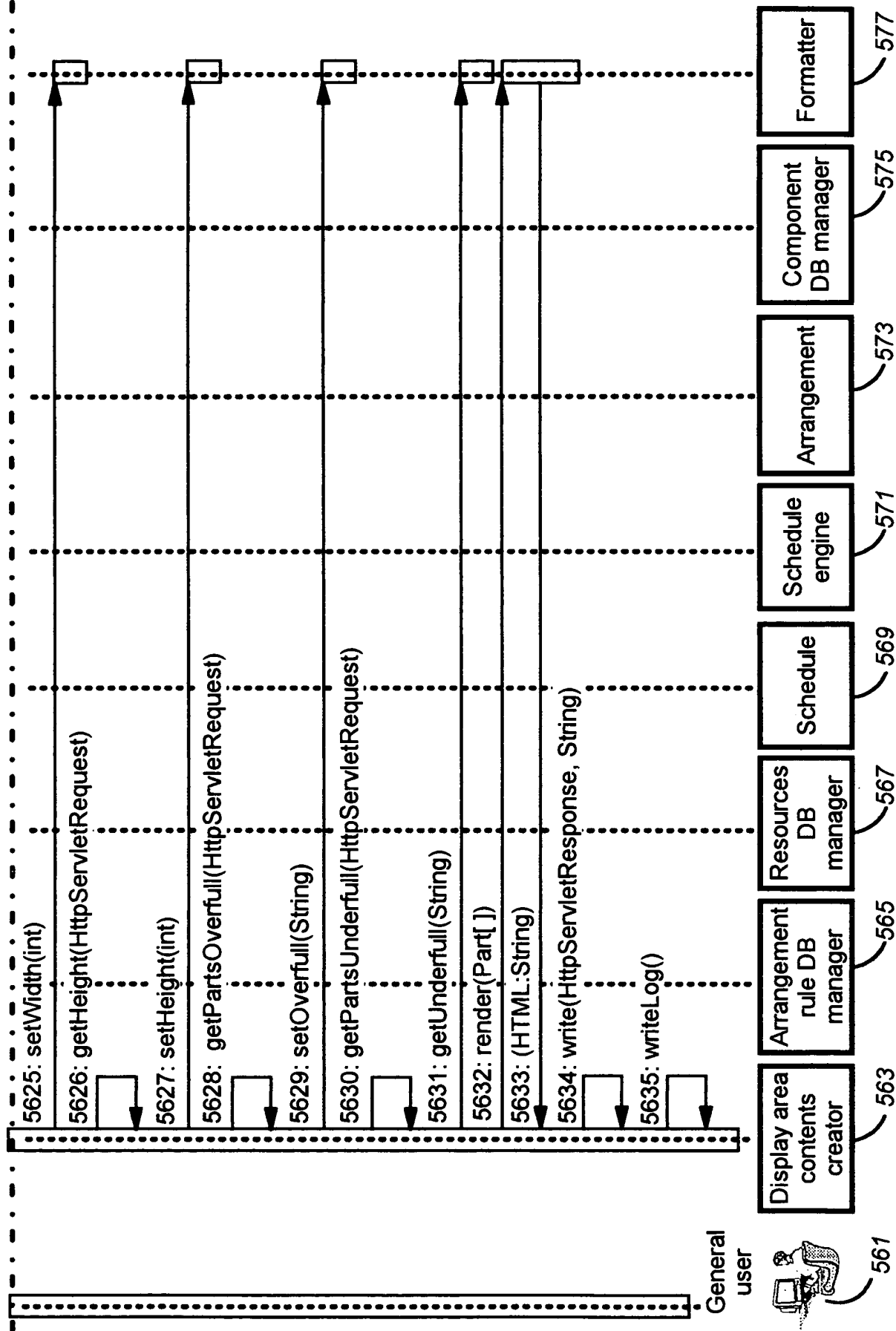


FIG. 19

```

<SERVLET code=icdacr codebase=servlet>
  <PARAM NAME="name" VALUE="product_ad">
  <PARAM NAME="style" VALUE="bannerFlow">
  <PARAM NAME="direction" VALUE="horizontal">
  <PARAM NAME="bgColor" VALUE="Gray"
  <PARAM NAME="width" VALUE="300">
  <PARAM NAME="height" VALUE="80">
  <PARAM NAME="position" VALUE="top">
  <PARAM NAME="default" VALUE="<IMG SRC='default_img.gif'>">
  <PARAM NAME="partsOverfull" VALUE="random">
  <PARAM NAME="partsUnderfull" VALUE="shrink">
</SERVLET>

```

701

703

Example of mounted display area 700

FIG. 20

Part ID	Part display area name	Part template ID	START	END	Publish
0001	product_ad	1001	1999-01-01 0:00	1999-07-01 0:00	1
0002	product_ad	1002	1999-07-01 0:00	2000-01-01 0:00	1
0003	product_inf	1003	1999-08-01 4:00	1999-09-20 23:00	0
.
.
.

Arrangement rule DB 720

FIG. 21

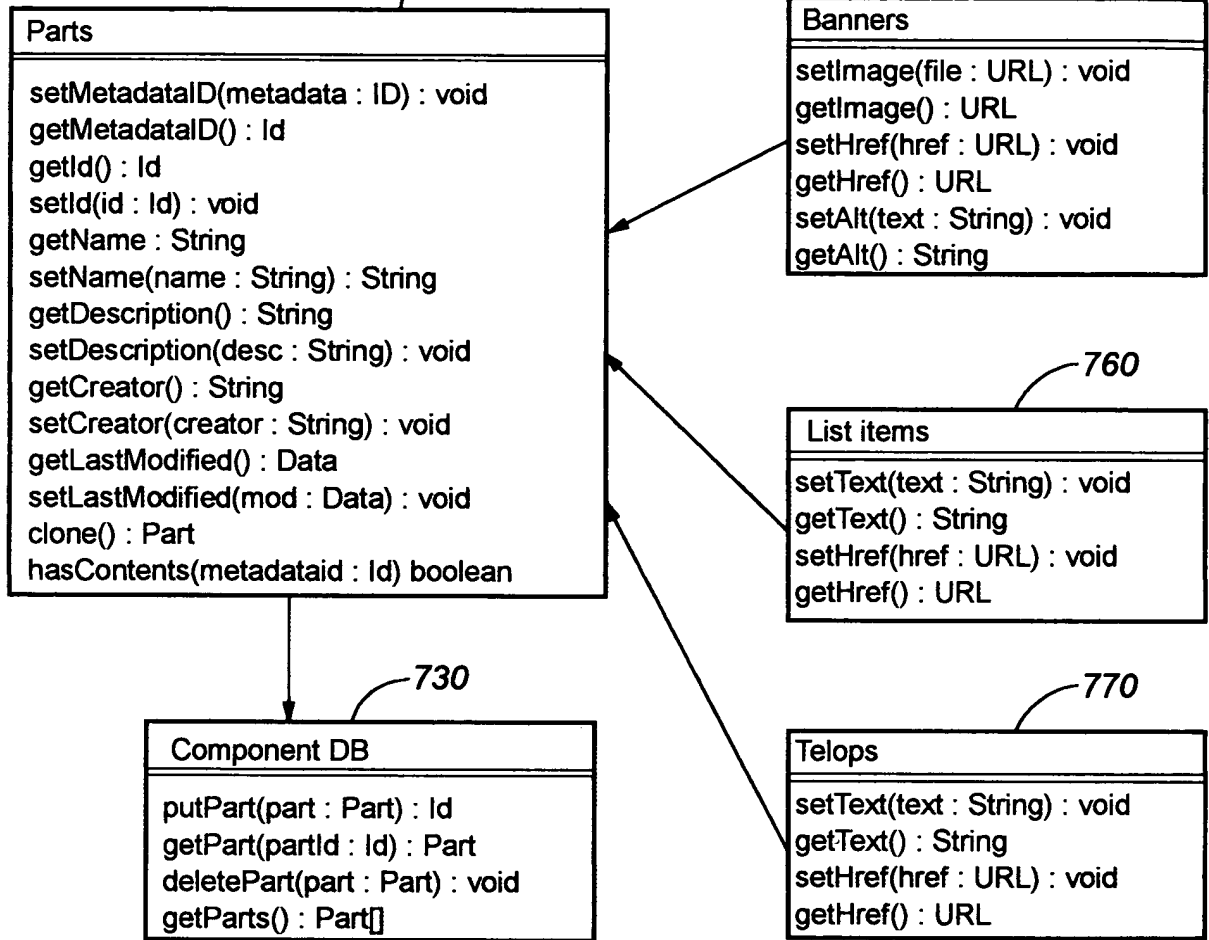


FIG. 23

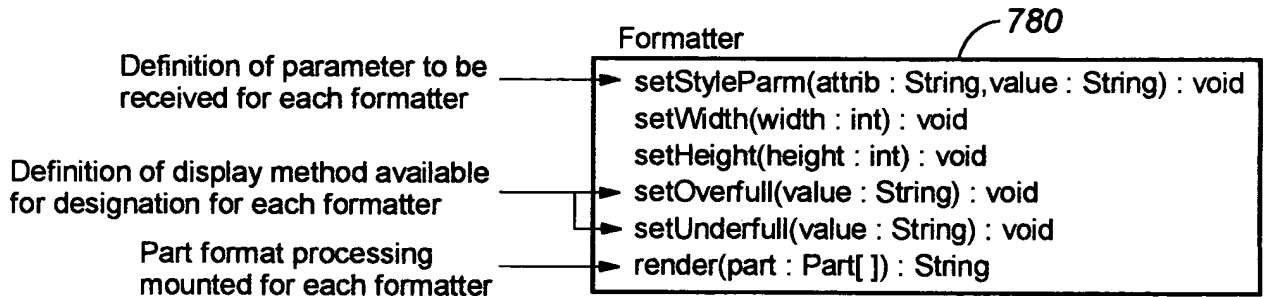


FIG. 22

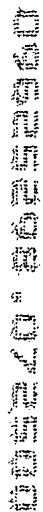


FIG. 24

Difference in format with display are style

```
<SERVLET code=indactr codebase=/servlet>  
  <PARAM NAME="name" VALUE="product_ad">  
  <PARAM NAME="style" VALUE="banner">  
  <PARAM NAME="width" VALUE="100">  
  <PARAM NAME="height" VALUE="40">  
  <PARAM NAME="position" VALUE="top">  
  <PARAM NAME="default" VALUE="<IMG SRC='default_img.gif'>">  
  <PARAM NAME="partsOverfull" VALUE="rotation=2">  
  <PARAM NAME="partUnderfull" VALUE="shrink">  
</SERVLET>
```

FIG. 25

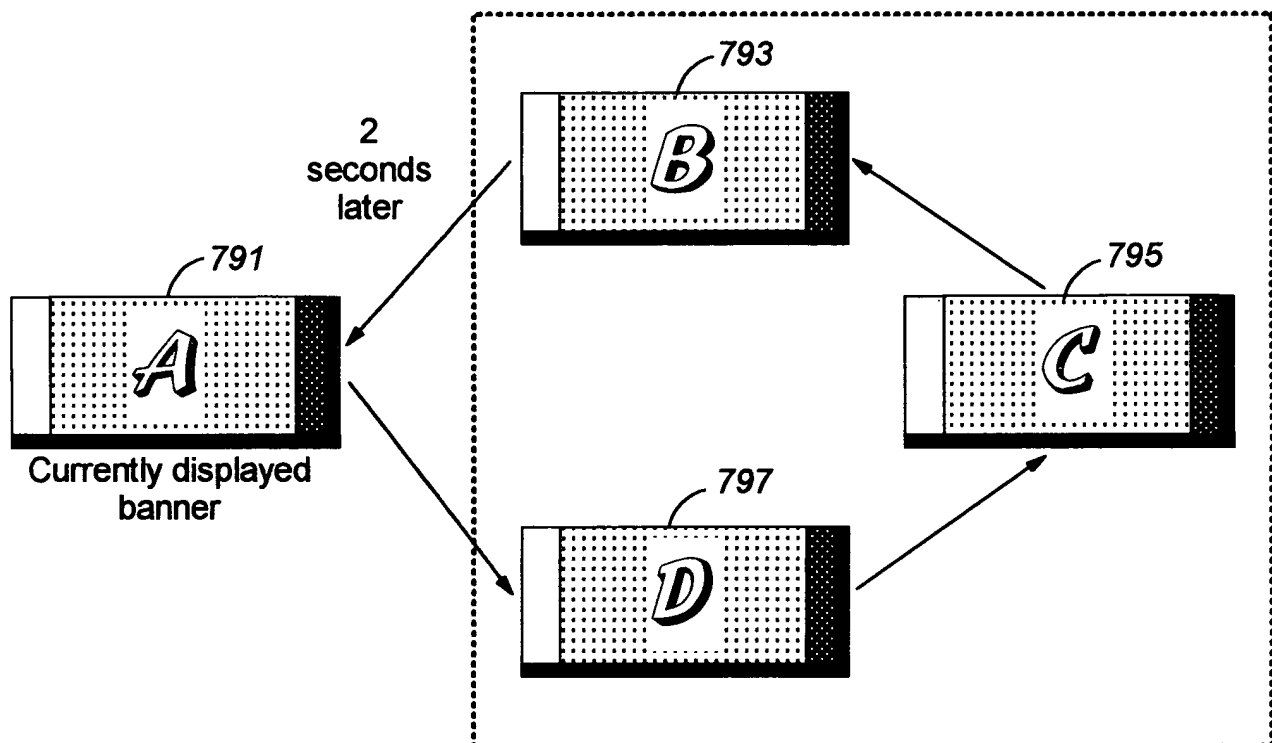


FIG. 26

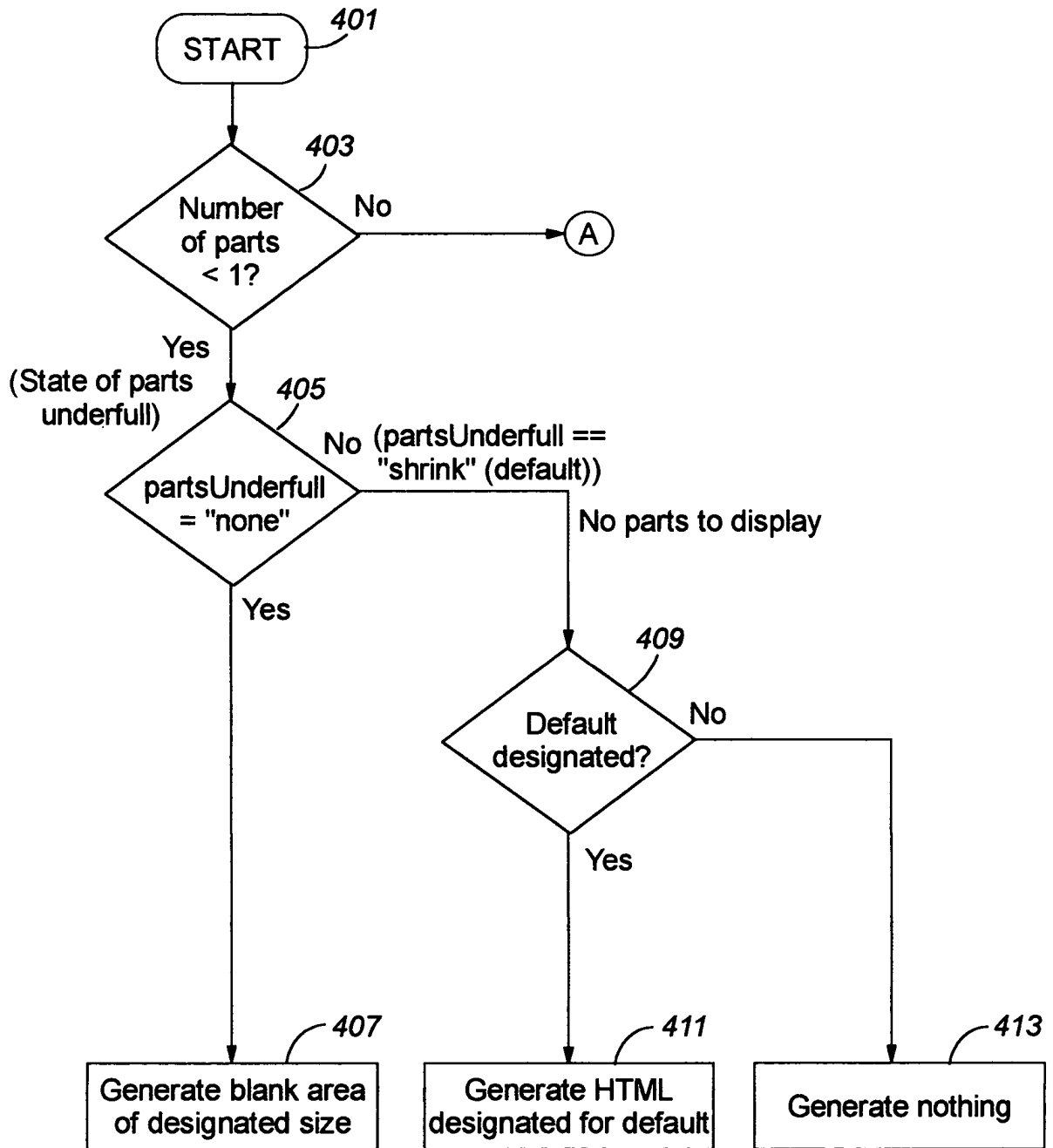


FIG. 27

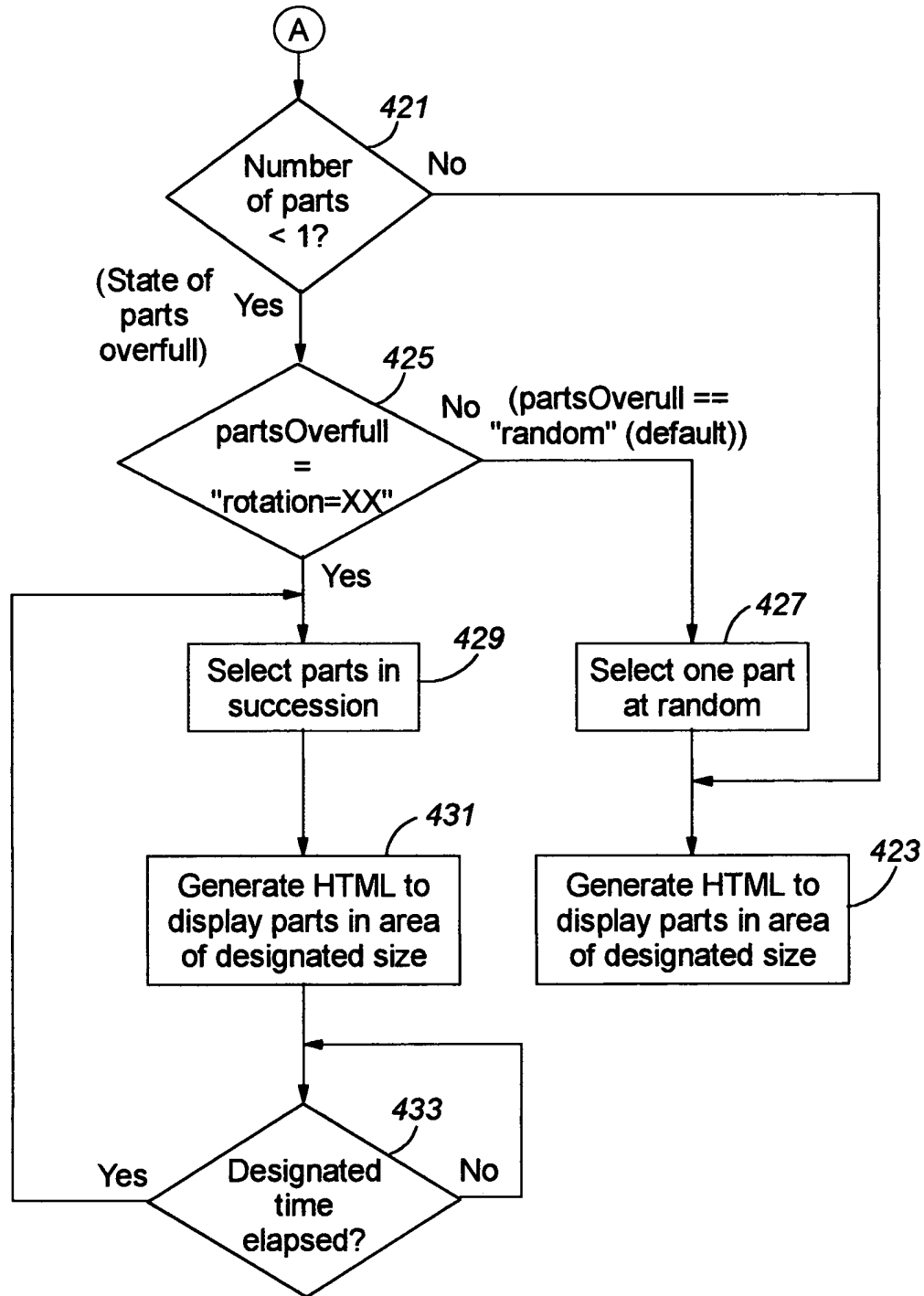


FIG. 28

```
<SERVLET code=indact codebase=/servlet>
  <PARAM NAME="name" VALUE="product_ad">
  <PARAM NAME="style" VALUE="bannerFlow">
  <PARAM NAME="direction" VALUE="horizontal">
  <PARAM NAME="bgColor" VALUE="gray">
  <PARAM NAME="width" VALUE="300">
  <PARAM NAME="height" VALUE="80">
  <PARAM NAME="position" VALUE="top">
  <PARAM NAME="default" VALUE="<IMG SRC='default_img.gif'>">
  <PARAM NAME="partsOverfull" VALUE="random">
  <PARAM NAME="partUnderfull" VALUE="shrink">
</SERVLET>
```

FIG. 29

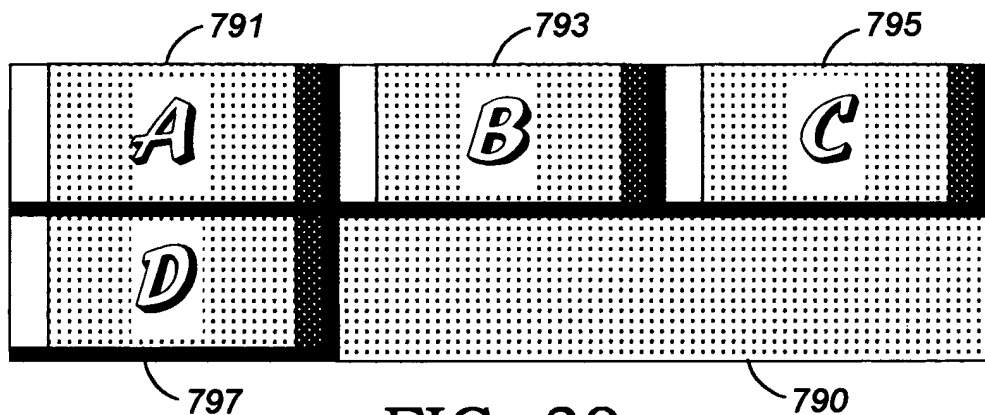


FIG. 30

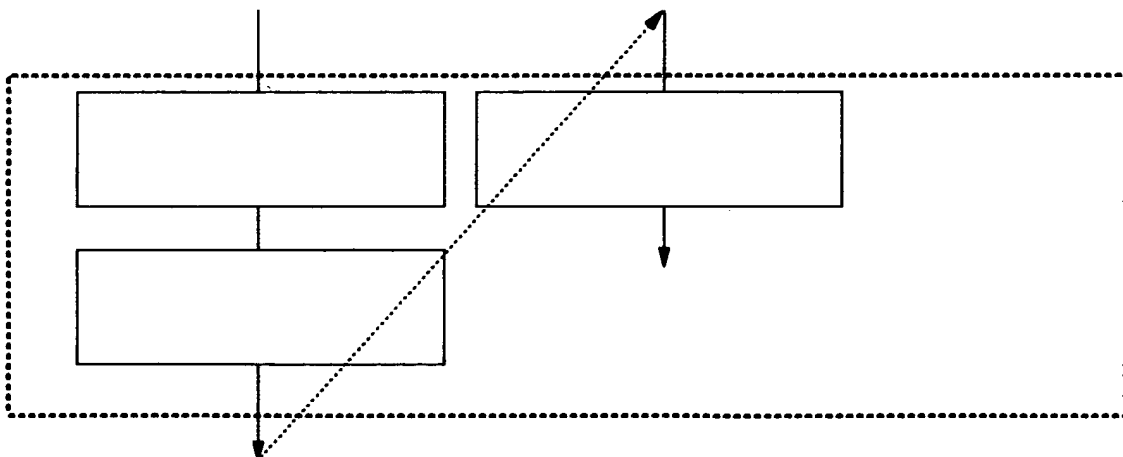


FIG. 31

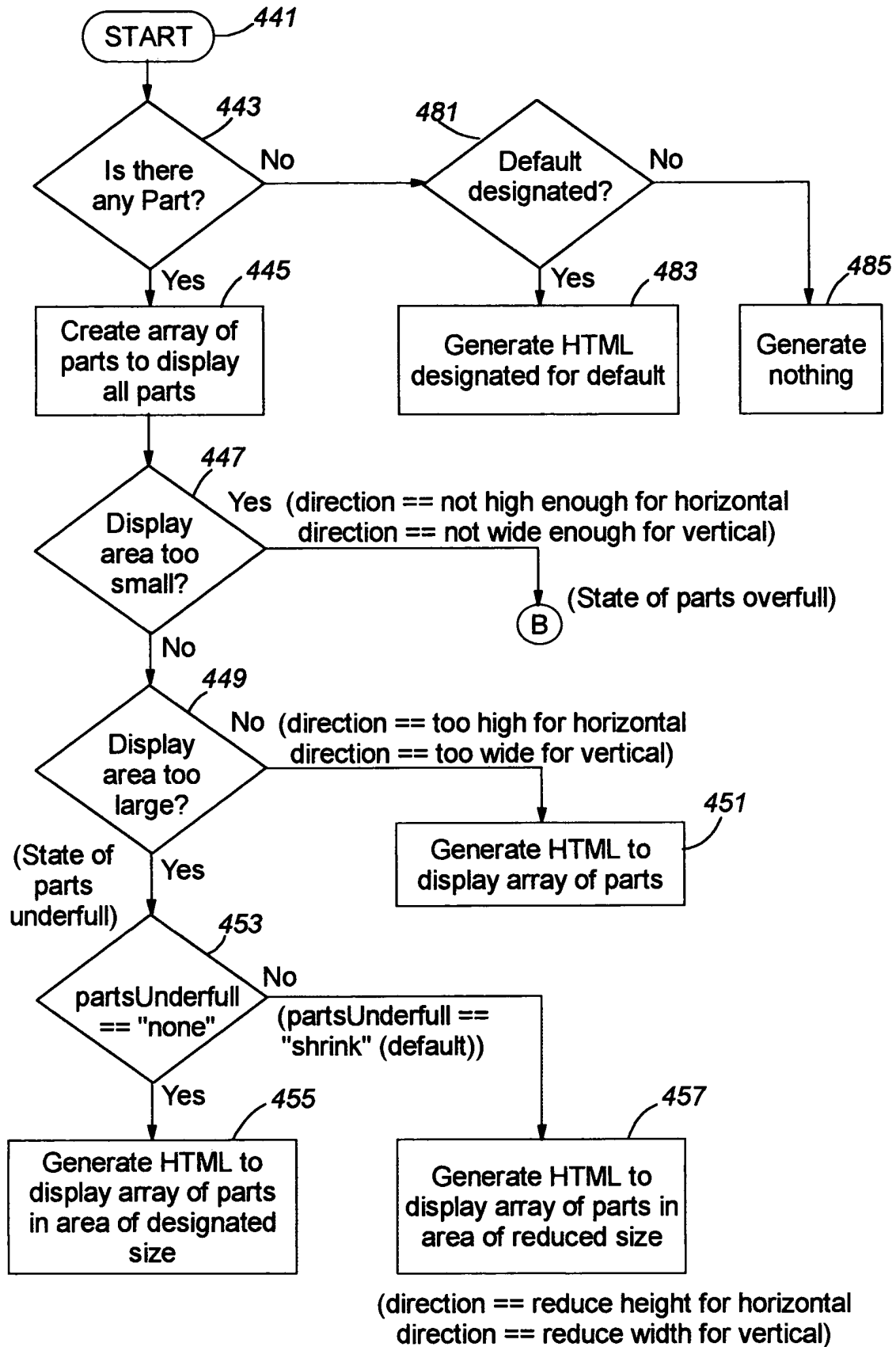


FIG. 32

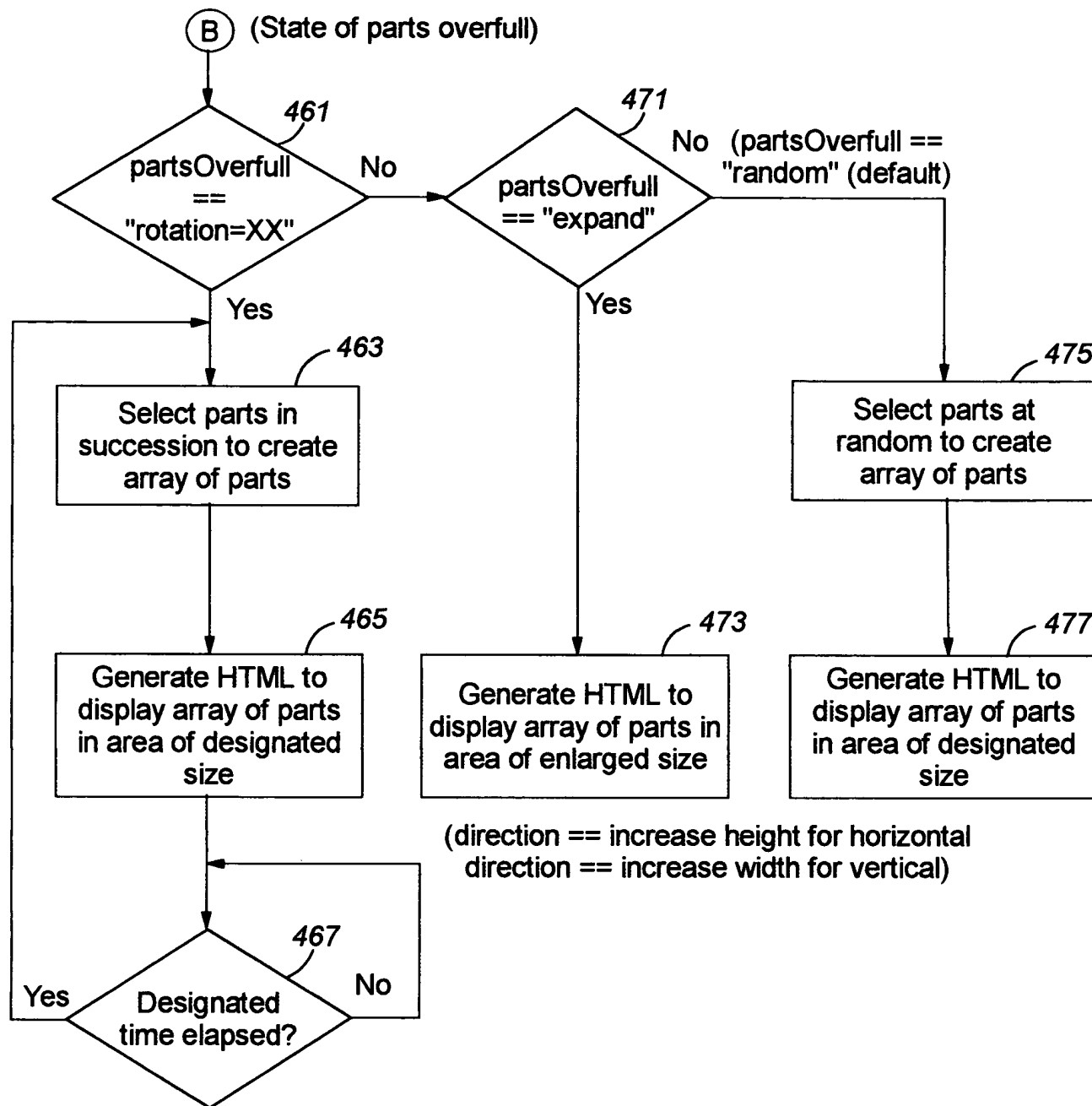
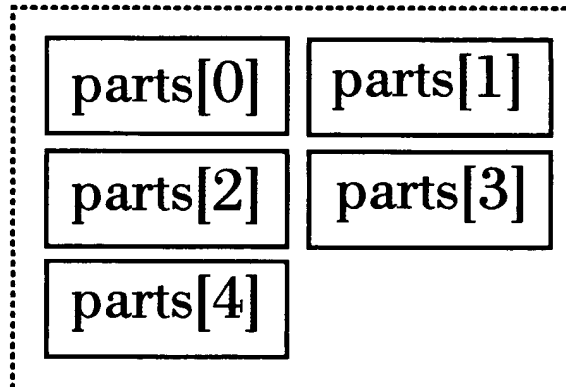


FIG. 33

Create array of parts to display all parts

parts[0...4]

direction == horizontal



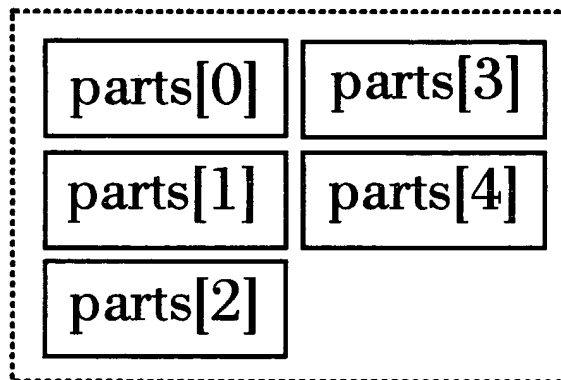
Parts arrangement

(0,0) = parts[0]; (0,1) = parts[1];

(1,0) = parts[2]; (1,1) = parts[3];

(2,0) = parts[4];

direction == vertical



Parts arrangement

(0,0) = parts[0]; (0,1) = parts[3];

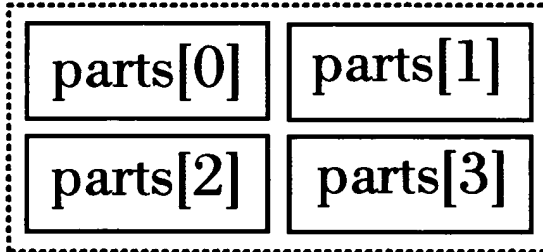
(1,0) = parts[1]; (1,1) = parts[4];

(2,0) = parts[2];

FIG. 34

Select parts in succession and create array of parts
parts[0...4]

Case of direction == horizontal

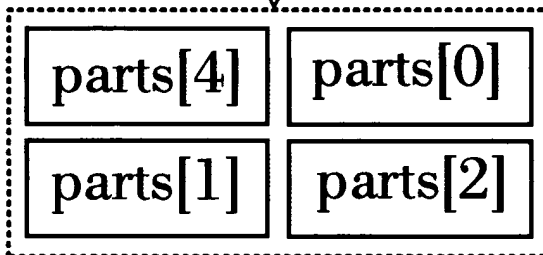


Parts arrangement

(0,0) = parts[0]; (0,1) = parts[1];

(1,0) = parts[2]; (1,1) = parts[3];

↓ After designated length of time



Parts arrangement

(0,0) = parts[4]; (0,1) = parts[0];

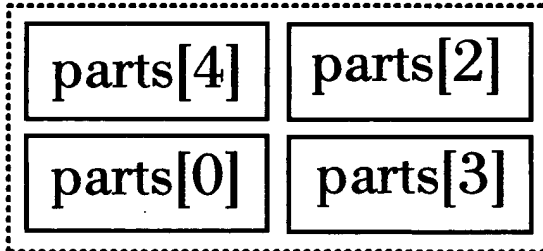
(1,0) = parts[1]; (1,1) = parts[2];

FIG. 35

Select parts at random and create array of parts
parts[0...4]

Select at random in sequence of parts[4], parts[2], parts[0], parts[3] and [arts[1]

Case of direction = horizontal



Parts arrangement

(0,0) = parts[4]; (0,1) = parts[2];

(1,0) = parts[0]; (1,1) = parts[3];

FIG. 36

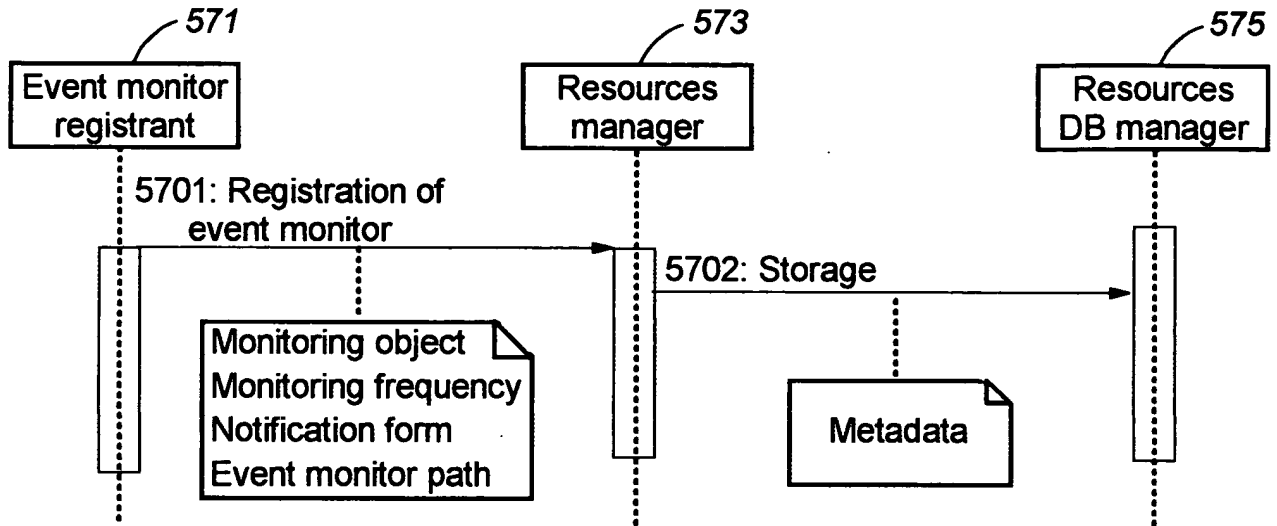


FIG. 37

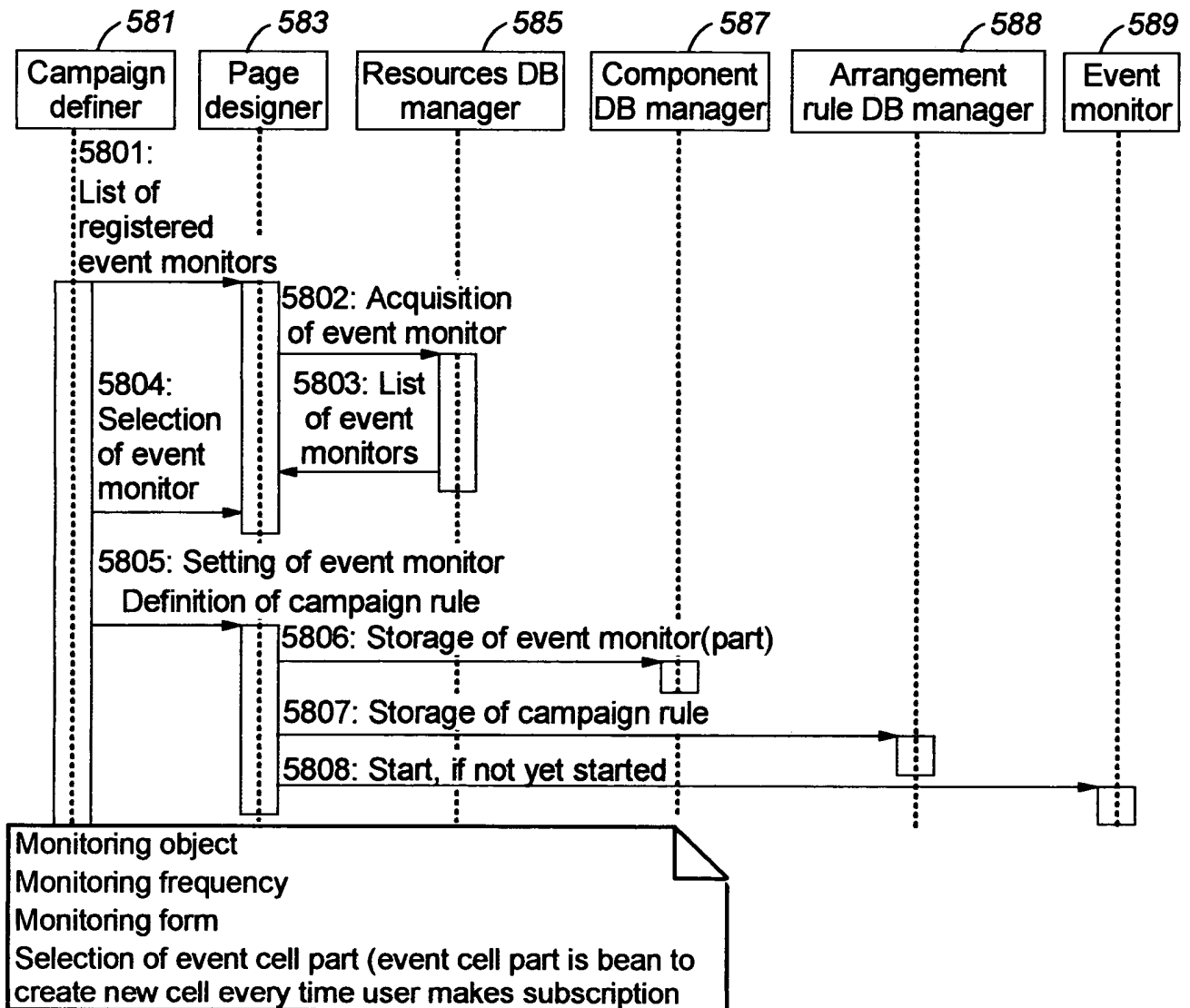


FIG. 38

731 Part type	721 Part ID	723 Part display area name	725 Page template ID	727 START	728 END	729 Publish	733 Event flag	735 Cell part ID
Banner	0001	product_ad	1001	1999-01-01 0:00	1999-07-01 0:00	1	1	2001
Banner	0002	product_ad	1002	1999-07-01 0:00	2000-01-01 0:00	1	1	2001
Telop	0003	product_inf	1003	1999-08-01 4:00	1999-09-20 23:00	0	0	2001
Event	0004						1	2003
.
.
.

720
Arrangement rule DB

FIG. 40

User ID	Cookie	E-mail address	User profile

720
User management DB

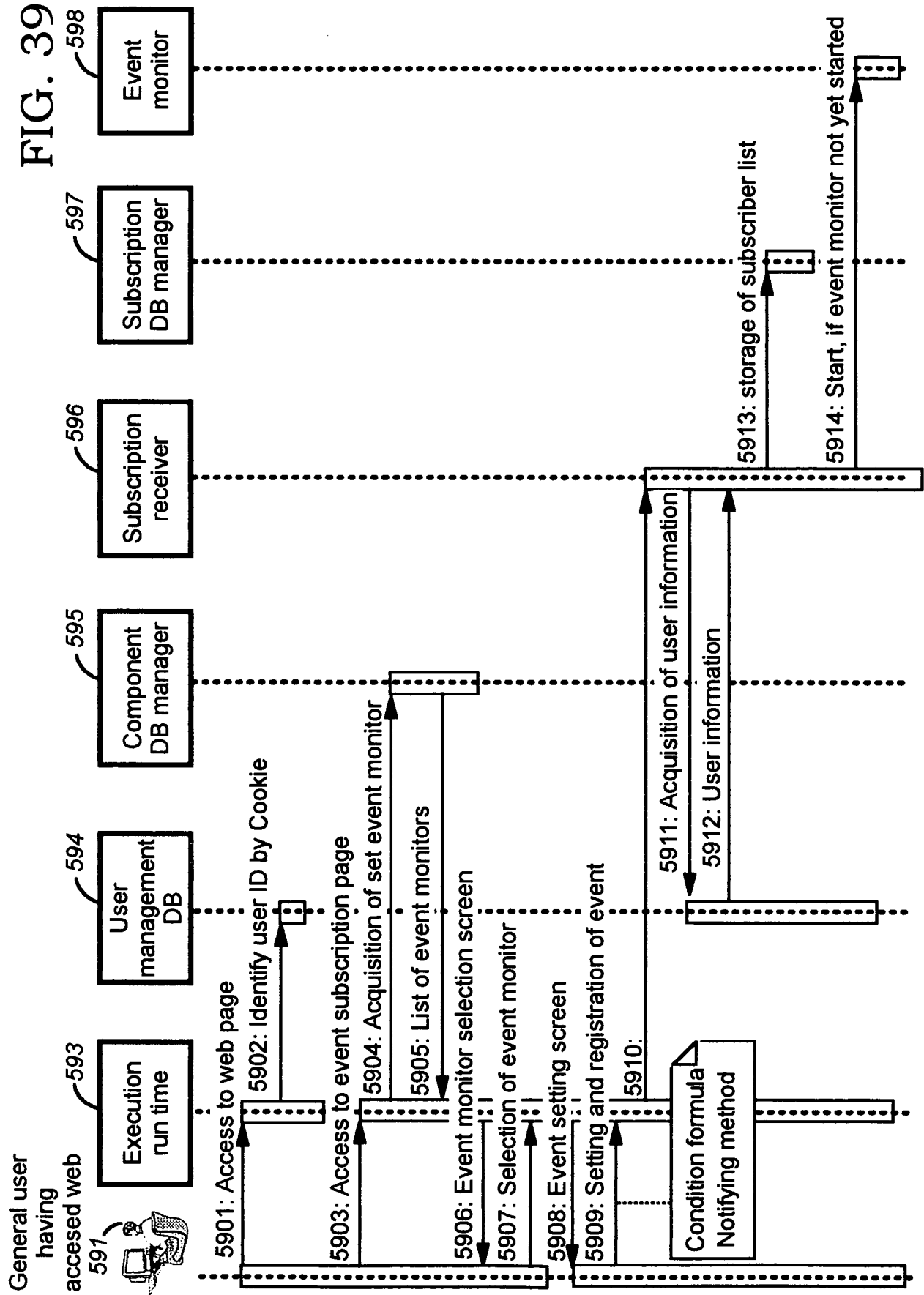


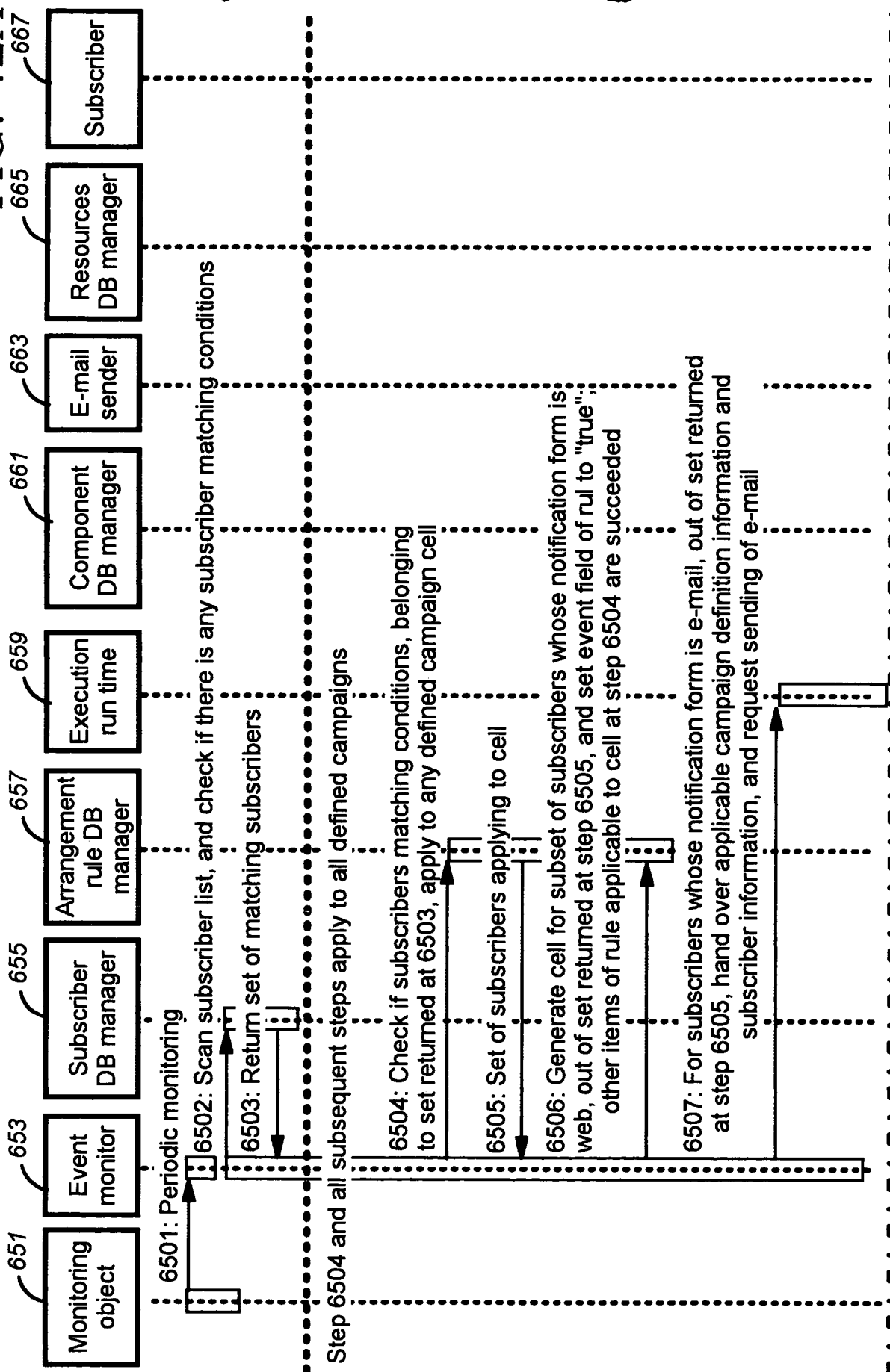
FIG. 41

Subscription ID	User ID	Part ID	Condition formula	Notification form	Campaign flag

750

Subscription DB

FIG. 42A



```
sequenceDiagram\n    participant System\n    participant User as User\n    participant Mailbox as Mailbox\n    participant Server as Server\n\n    Note over System: 6508: Acquire e-mail sender designated by campaign definition\n    System->>User: 6509: Retrieve\n    activate User\n    User-->>System: 6510:\n    deactivate User\n    Note over System: 6511: E-mail sender returns\n    System->>Mailbox: \n    activate Mailbox\n    Mailbox-->>System: \n    deactivate Mailbox\n    Note over System: 6512: Request sending of e-mail based on subscriber information\n    System->>Server: \n    activate Server\n    Server->>System: 6513: Acquire text and sender information designated at setting\n    deactivate Server\n    Note over System: 6514: Text and sender information\n    System->>Mailbox: \n    activate Mailbox\n    Mailbox-->>System: \n    deactivate Mailbox\n    Note over System: 6515: Send e-mail\n    System->>User: \n    activate User\n    User-->>System: \n    deactivate User
```

The sequence diagram illustrates the process of acquiring and sending an email based on a campaign definition. The participants involved are the System, User, Mailbox, and Server.

- 6508:** The System initiates the process by acquiring the email sender designated by the campaign definition.
- 6509:** The System requests the User to retrieve the sender information.
- 6510:** The User returns the retrieved sender information to the System.
- 6511:** The System sends the email sender information back to the Mailbox.
- 6512:** The System requests the sending of an email based on subscriber information from the Mailbox.
- 6513:** The System acquires the text and sender information designated at the setting from the Server.
- 6514:** The System sends the acquired text and sender information back to the Mailbox.
- 6515:** The System sends the email to the User.

FIG. 43

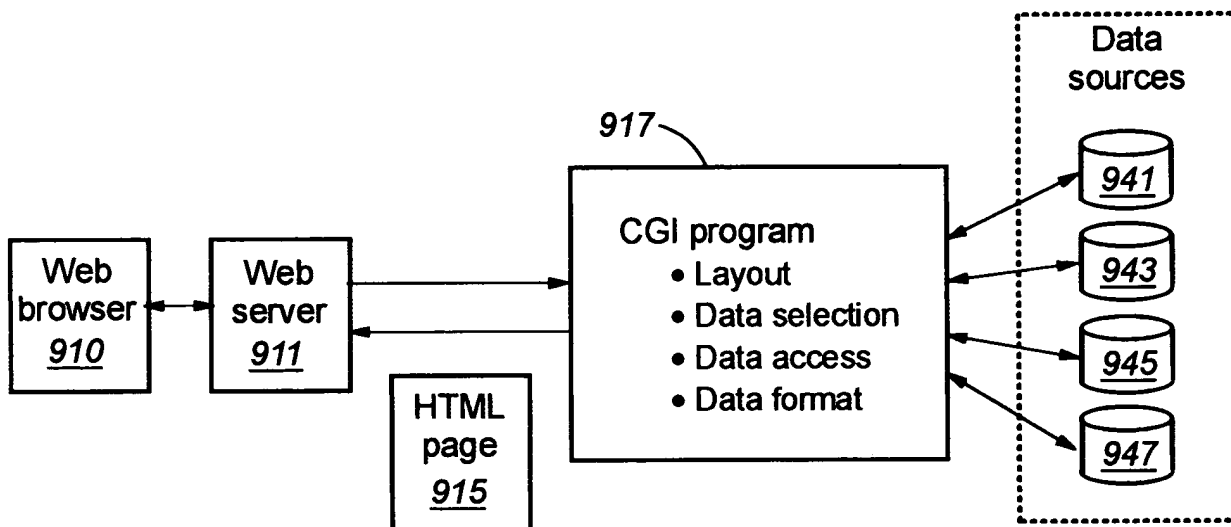


FIG. 44

